

System Administration of a C-TOOLS Server

0. About this Document

Table of Contents

1. Introduction
2. Comments on Installing
3. Checklist of Functionality
4. Backing up
5. Maintenance
6. Miscellaneous Notes

License

As indicated in each software file, those items of C-TOOLS project funded by the NSF are licensed under the terms of the GNU General Public License Version 2.0 (<http://www.gnu.org/copyleft/gpl.html>).

Disclaimer

The C-TOOLS software is made available 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.

1. Introduction

This is a general guide to C-TOOLS and an overview of system administration tasks. The goal of this document is to describe tasks a system administrator may need to perform on a C-TOOLS server. For more information, consult some of the README files inside the source tree, especially build/README. During the course of administering a C-TOOLS server, system administrators will need to execute routine linux commands. Prior experience with linux-type systems (e.g. RedHat Linux, Debian Linux, FreeBSD, or Mac OS/X) is recommended. Expertise should be comparable with knowledge sets such as those listed by the Linux Professional Institute (http://www.lpi.org/en/obj_201.html).

C-TOOLS is educational software that delivers a java applet over the world-wide web to support concept mapping. Students develop concept maps in an automated classroom. A significant feature of the C-TOOLS software is the Robograder™. The robograder helps an instructor iteratively develop and apply an artificially intelligent rubric to a concept map.

Because a C-TOOLS server is used for the delivery of online learning solutions, added care needs to be devoted to the maintenance of a C-TOOLS server. Improper functioning of the server can result in confusing and sometimes

irreconcilable situations for students, instructors, and educational research investigators. The developing nature of concept mapping learning online, the open nature of the C-TOOLS software and the usage of middleware all provide a good opportunity for novel customizations should you be able to expend that kind of effort once you have a C-TOOLS server running.

Also, please note that filesystem paths in this document may vary from the actual installation of C-TOOLS software on your system.

2. Comments on Installing

In order to install the C-TOOLS software, two tasks must be performed: the operating system must be properly configured, and then the A-A-P tool is used to perform the installation. For specific information on installing C-TOOLS, please consult the C-TOOLS installation instructions document.

Most filesystem parameters can be altered with command-line arguments passed to the A-A-P installation/upgrade mechanism. Note that A-A-P is used instead of a direct Makefile strategy or an autoconf/automake strategy. A-A-P is simple, modern, and, for purposes of cross-platform installation, only has one significant software dependency, python.

A more complete description of installing C-TOOLS is described in build/README. The build/README document also helps describe the variety of command-line options that can configure software installation parameters. To run the installation with the A-A-P tool:

```
cd build ... go to the build directory inside C-TOOLS' source tree
aap build PREFIX=null ... (should be run as root)
aap install PREFIX=/ ... (should be run as root)
```

You can also update a machine remotely, the following syntax will work with A-A-P (replace ctools.msu.edu with the appropriate ip name of your server machine):

```
aap install PREFIX=scp://root@ctools.msu.edu//
```

From the context of production-line usage, there should never be a need for deinstalling C-TOOLS from the server. There is likely nothing a deinstallation will do to improve subsequent C-TOOLS software installations.

To deinstall: go to the build directory inside C-TOOLS' source tree and use the "aap deinstall" command (should be run as root). Deinstallation tries to preserve user data and system configuration information. A backup of user data should be performed prior to any software deinstallation.

3. Checklist of Functionality

After setting up a C-TOOLS software and dependencies, here is a checklist of functionality tests that should be run prior to using the C-TOOLS server in the classroom. While the complexity of the overall set of C-TOOLS software and dependencies may not be simple to address, successful testing of this checklist would generally indicate a high degree of confidence in a working C-TOOLS server. The steps in this checklist should be performed manually by a human operator. The goal is to prevent easy-to-catch glitches from reaching an environment of classroom usage.

3.1 Classroom Environment

Does the "Submit" assignment applet button work?
Does the "Submit" link from the Map Gallery interface work?
Does the "Create/Edit" assignment functionality work?
Does browsing through assignment submissions (navigate->enter data navigator) work?
Are assignments successfully posted?

3.2 Web page navigation

Can a new user register?
Can a new user register as an existing user?
Can an existing user log in?
Does ctools.html refresh to the front page?
Does SSL/non-SSL switching (the "Standard" and "Secure" modes) work?
Does "Map Gallery" presentation work?
Does "Map Gallery" "browse previous versions" work?
Does "Map Gallery" "edit this map" work?
Does specifying a title from the "Map Gallery" work?
Does "Map Gallery" "delete this map" work?
Does "Map Gallery" "recovering from trash folder" work?
Does "Logout" work for known users?
Does "Forgotten Password" work for anonymous users (including the e-mail message that gets sent)?
Does "Change password" work for known users?
Do the generated web pages conform to the web_page_standards.txt document?

3.3 Applet testing

Does "Move" work?
Does "Add Concept Word" work?
Does "Add Linking Word" work?
Does "Add Connecting Link" work?
Does "Delete Word" work?
Does "Delete Lines" work?

- Do the "VIEW" features work?
- Does "Toggle Color" work?
- Is toggled color persistent for "Print Previews"?
- Does "Save Now" work from the front page (for anonymous user)?
- Does "Save Now" work from a logged in session?
- Does "Show Robograder" work?

3.4 Log Entries

- Are normal log entries occurring inside WEBSERVERLOGS/ctools_log for each of the actions shown above?
- Are corresponding log entries occurring inside WEBSERVERLOGS/httpd-access_log for much of the above?
- Are any suspicious log entries occurring inside WEBSERVERLOGS/httpd-error_log during the testing?

3.5 Processes

Are the CmapImageSpoolProcessor.pl, DaemonToGenerateImageSpool.pl, and DaemonToCleanCtoolsTmp.pl processes all running (as started by /usr/local/etc/rc.d/ctools_control.sh)?

Is mysqld running?

Is zope running?

3.6 Dynamic Directories

Are the subversion-managed user directories all operating as expected?

4. Backup

The command to back up a C-TOOLS server is /usr/local/ctools/bin/ctools_backup (use perldoc for more information). This script does not require any command-line options to operate normally.

If and when you need to restore backed up data to a C-TOOLS server, use the command /usr/local/ctools/bin/ctools_restore (use perldoc for more information). Typical invocation of ctools_restore is to pass it a timestamp as a command-line argument. The timestamp should correspond to a directory name inside /home/adminCTOOLS/backups/. Typically, these directory names are labelled in the format: YYYYMMDDhhmmss. For example, perl /usr/local/ctools/bin/ctools_restore 20030120040418.

Both ctools_restore and ctools_backup need to be run as the adminCTOOLS user.

System administrators may also want to back up those parts of the C-TOOLS installation that they customize.

5. Maintenance

Also, C-TOOLS relies on a variety of databases and database transaction utilities (e.g. MySQL, Berkeley-DB, Subversion). The command to verify integrity and operation of C-TOOLS database services is `/usr/local/ctools/bin/ctools_users_directory_check` (use `perldoc` for more information). An example invocation is `ctools_users_directory_check --nodaemon --outfile=/usr/local/ctools/log/dir_check.log`. The `ctools_users_directory_check` tool should be run as the `www` user (or whatever the `$WEBUSER` is set to during installation time). Since this script takes a while to run, you may wish to pass it the `--verbose` flag so that you can monitor progress.

Also, disk space frequently fills up (every 6 months or so) due to the accumulation of subversion log files. For now, a system administrator needs to remove these log files about once a semester. Be sure to consult the subversion manual on how to do this. An example set of commands is shown below (these commands may take a while, say 10-20 minutes). Before performing these commands, be sure to verify your user directories, and backup user data with `ctools_backup`. Note: you may want to do "xargs -l X svnadmin list-unused-dblogs X" on FreeBSD systems.

```
cd /usr/local/www/data/ctools/users
find . -type d -name svnrepos | xargs svnadmin list-unused-dblogs | xargs rm
```

There are more advanced ways to ensure that these log files do not accumulate. Also, the newer versions of subversion and their default data formatting should not have this problem.

It is also important to maintain the operating system of a C-TOOLS server. You should follow instructions for that operating system in order to update software packages (e.g. rpm or BSD's port system), maintain a firewall (e.g. iptables), restrict services (e.g. `/sbin/chkconfig`, `/sbin/service`), manage disk space (e.g. `df`, `mrtg`), and any other regular tasks for proper system administration of your C-TOOLS server's operating system.

6. Miscellaneous

A common task when first setting up a C-TOOLS server is to customize the logos that appear on the front ("splash") pages, soft-links are used. For example, `/usr/local/www/data/ctools/icons/leftlogo.png` can be made to point to a file called `blockMM.png` if one is from "Maritime Memorial University" (and `blockMM.png` is

that school's logo). The C-TOOLS production server currently uses Spartan block "S" logo on the left, and an NSF logo on the right. Here is an example set of commands to control this kind of customization:

```
cp /somewhere/blockS.png /usr/local/www/data/ctools/icons/.  
cp /somewhere/nsf.png /usr/local/www/data/ctools/icons/.  
cd /usr/local/www/data/ctools/icons; ln -fs nsf.png rightlogo.png  
cd /usr/local/www/data/ctools/icons; ln -fs blockS.png leftlogo.png
```

There can be a wide range of institution-specific possibilities when it comes to customizing a C-TOOLS distribution (adding your school colors), integrating a C-TOOLS distribution (synchronizing it with your institution's registrar database, user authentication, or integrating it with preferred course management software such as Angel or Blackboard), and various configurations (receiving status reports from the machine, receiving messages from users, and filesystem management strategies).

Authentication and database-level configurations can be set by adjusting the relevant C-TOOLS modules (usually located inside `/usr/local/ctools/lib/`).

Customizing the appearance of a C-TOOLS distribution involves altering stylesheets (such as `ctoolshelp.css`) and various template files which specify headers, footers, and layout formats for the web pages.

More configuration of a C-TOOLS server can be adjusted post-installation by visiting the `/usr/local/ctools/etc/` directory. Filenames inside this directory are fairly intuitive (e.g. `bottombar.htmlfrag`, `ctools.conf`, and `assignment_privileges`).

To perform basic diagnostics and configuration of the operating system, visit the `ctoolrepos/trunk/system_dependencies/tests` directory, look at the README file. There are various diagnostic scripts (e.g. `package_requirements.pl` and `mysql_requirements.pl`). These scripts were developed for RedHat operating systems as opposed to Mac systems, but some of their contents may be useful to look at by the system administrator.

The suggested order of execution for basic operating system configuration scripts (present inside `system_dependencies/tests`) is:

1. `package_requirements.pl`
2. `mysql_requirements.pl`
3. `dev_requirements.pl`
4. `perl_requirements.pl`

Although you may see a few notes and warning messages, the only thing that indicates a need to reconfigure your system are the `**** ERROR ****` messages.

Also, here is a simple approach for diff-ing production filesystem against code base should you be branching or upgrading the C-TOOLS code base.

```
sudo aap install PREFIX=/somepath/YYYYMMDD_ctools
cd /somepath/YYYYMMDD_ctools
diff -r --brief . / | grep -v 'Only in /'
```

Here is a more complicated approach for diff-ing production filesystem against code base.

```
sudo aap install PREFIX=/somepath/YYYYMMDD_ctools
cd /somepath
tar czvf YYYYMMDD_ctools.tar.gz YYYYMMDD_ctools
(transport YYYYMMDD_ctools.tar.gz to server to be verified)
find YYYYMMDD_ctools | cut -b16- | perl -nle 'if (-e $_) {print $_ unless -d $_;}' |
tar -I - -c -f - | tar xCf YYYYMMDD_existing -
diff --brief -r YYYYMMDD_existing YYYYMMDD_ctools
```