The CoTeia Guide

Marco Aurélio Graciotto Silva

The CoTeia Guide

by Marco Aurélio Graciotto Silva

This document provides installation instructions for CoTeia, an colaborative web editor (also known as wiki).

Table of Contents

List of Examples

Chapter 1. Installing

1.1. Introduction

CoTeia can be installed in a computer running Windows or Linux (or any POSIX compatible) operational system. The recommended plataform is Linux and specific instructions for installing under the latest stable Debian or Conectiva GNU/Linux distributions are available. Although Windows is supported, some of the latest features may lag behind: these will be documented within this document.

1.2. Requirements

There are listed below the required software to run the CoTeia. Each item is identified with the URL for its home-page, the minimum version required and the version tested by the developers (and, therefore, recommended). Whenever a version is written with an "x", like "2.0.x", the letter "x" means any possible number. For example, "2.0.1" or "2.0.22" would be valid versions.

- CoTeia (http://incubadora.fapesp.br/projects/coteia/) (1.2.0/1.3.0).
- Apache HTTPd Server (http://httpd.apache.org/) (1.3.x/2.0.x).
- PHP (http://www.php.net/) (4.1.x/4.3.x).
- CVS (http://www.cvshome.org/) (1.10.x1.11.x), recommended for Linux or Unix, or CVSnt (http://www.cvsnt.org) (2.0.x2.0.x), recommended for Windows.
- MySQL (http://www.mysql.com/) (3.23.x/4.0.x)
- *(optional)*Java Runtime Environment 2 (http://java.sun.com) (1.2.x/1.4.x)

> **Important:** The Java Runtime Environment is used for XML transformation (XSLT) when PHP hasn't support for XSLT enabled.

1.3. Configuration

1.3.1. Configuring the Apache HTTPd

The HTTPd server's hostname and listening port are needed to configure CoTeia. These information can be gathered from `httpd.conf`, usually installed into the Apache's server root.

The server’s hostname must be a valid Internet name (e.g. www.google.com) or the machine IP’s address (e.g. 200.250.8.1). The parameter "ServerName" holds the default HTTP server’s name:

```
#
# ServerName allows you to set a host name which is sent back to clients for
# your server if it’s different than the one the program would get (i.e., use
# "www" instead of the host’s real name).
#
# Note: You cannot just invent host names and hope they work. The name you
# define here must be a valid DNS name for your host. If you don’t understand
# this, ask your network administrator.
# If your host doesn’t have a registered DNS name, enter its IP address here.
# You will have to access it by its address (e.g., http://123.45.67.89/)
# anyway, and this will make redirections work in a sensible way.
#
# 127.0.0.1 is the TCP/IP local loop-back address, often named localhost. Your
# machine always knows itself by this address. If you use Apache strictly for
# local testing and development, you may use 127.0.0.1 as the server name.
#
ServerName 127.0.0.1
```

From the excert above, the "ServerName" is "127.0.0.1" (a IP address is usually set when the machine haven’t got a valid Internet name).

> **Important:** The "ServerName" maybe not work. Your server may be configured to service virtual hosts. Contact the server admin for more information.

The listening port usually is 80 when Apache is a site-wide installation, and 8080 when installed by the user. This setting is defined by the parameter "Listen":

```
#
#Port: The port to which the standalone server listens.  Certain firewall
#products must be configured before Apache can listen to a specific port.
#Other running httpd servers will also interfere with this port.  Disable
#all firewall, security, and other services if you encounter problems.
#To help diagnose problems use the Windows NT command NETSTAT -a
#
Listem 80
```

Finally, it’s necessary the server’s root. This directory is the one the HTTP server search for files, and is set by the variable "DocumentRoot":

```
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "/srv/www"
```

By now, you should know the server's name, port and document root. If the information is correct, you probably can access the URL "http://servername:listeningport/" and the result will be the contents of the file `index.html` (or similar) that can be found at the documentroot.

1.3.2. Configuring the PHP

The current version of CoTeia requires the activation of the PHP's parameter "register_globals". The latest PHP's version disables it by default, so it's neeeded to activate it, setting the variable at `php.ini`.

> **Tip:** This file is usually found at `/etc` in Linux and Unix systems.

The except below shows the default setting for "register_globals":

```
; You should do your best to write your scripts so that they do not require
; register_globals to be on;  Using form variables as globals can easily lead
; to possible security problems, if the code is not very well thought of.
register_globals = OFF
```

Please change the parameter to "ON".

1.3.3. Configuring the Mysql

MySQL usually doesn't require any special configuration. You just need the server's hostname, username, password and database's name. Ask the database admin to create an username and set up a database for it.

> **Tip:** If you are the system admin, the installation scripts can setup this automatically for you.

1.3.4. Configuring the CVS

The simplest use of CVS is by creating a local repository. This is accomplished automatically by CoTeia's setup tool. This is the fastest and easiest solution. But, if desired, any other access method can be employed (e.g., access, using the pserver method, a remote repository).

This guide doesn't cover the configuration of a remove CVS server. Ask your system admin for a CVS repository that can be used by CoTeia. The following CVS server data will be necessary: hostname (or IP address), access method (usually "pserver"), module, and, optionally, username and password.

1.3.5. Configuring the CoTeia

The only file that needs modification is `config.php`. Every parameter that must be set has information about its value. There is a `config.php.eg` in the directory `doc/` that can be used as a base configuration, just copy it to `config.php` and edit the unfilled parameters.

Configurable parameters

CVS_METHOD

The method that will be used to access the CVS repository. Possible values are *local* and *pserver*.

Example 1-1. Setting *pserver* as CVS access method

```
$CVS_METHOD = "pserver";
```

CVS_SERVER

Computer name or IP address that hosts the CVS repository. Mandatory when using the *pserver* method, ignored for the *local* method.

Example 1-2. Setting the local machine as CVS server

```
$CVS_SERVER = "localhost";
```

CVS_ROOT

The repository path at CVS's server. That's usually informed your system admin when accessing a remove repository with the method *pserver* . When using the method *local*, it's the directory that holds the cvs repository. Such directory must be visible (and writable) by the httpd daemon.

Example 1-3. Setting the cvsroot to a local directory, inside the CoTeia's tree

```
$CVS_ROOT = "$PATH_COWEB/local_cvs";
```

CVS_MODULE

A CVS repository is organized into modules, usually with individual access policies. This parameter holds the module name that CoTeia will have access (full access, writing and reading). Your system admin must have informed you this data. If using an local repository, this is a subdirectory within the *CVS_ROOT* (the setup script will automatically create it).

Example 1-4. Setting the module to "html"

```
$CVS_MODULE = "html";
```

CVS_USERNAME

Example 1-5.

```
$CVS_USERNAME = "";
```

CVS_PASSWORD

Example 1-6.

```
$CVS_PASSWORD = "";
```

CVS_CHECKOUT_DIR

Example 1-7.

```
$CVS_CHECKOUT_DIR = $PATH_COWEB . "/repository";
```

CVS_PASSFILE

Example 1-8.

```
$CVS_PASSFILE = $CVS_CHECKOUT_DIR . "/.cvspass";
```

Example 1-9.

```
$DEFAULT_USER = "";
```

CoTeia depends heavyly on `config.php`. Please, check the file correctness running "php -l `config.php`".

After setting the `config.php`, run tun the setup script (`tables/setup.php`):

```
php tables/setup.php
```

Next, the SQL scripts (compiled by `setup.php`) should be run againt the MySQL. The files are:

1. coteia.sql
2. chatserver.sql
3. coweb-users.sql
4. eclass.sql
5. groupnode.sql

You can run the files using the console MySQL client. Use the "\." statement followed by the filename:

```
# mysql -h [mysql_hostname] -u [username] -p

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 4.0.18-nt

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

Mysql>

Mysql> \. coteia.sql;
```

Important: The SQL scripts may require some modifications, especially about the username's creation.

1.4. Upgrading

When upgrading from an old CoTeia, there’re several factors that must be checked: CVS, new transformation stylesheet, etc. The easiest solution is:

1. Entirely remove the directories $CVS_CHECKOUT_DIR, $PATH_ARQUIVOS, $PATH_XHTML and $PATH_XML.
2. Run `setup.php` (the setup script).

The first access to every page will be slower as the CVS and HTML page will be compiled, but the procedure is much easier than rebuild every file again (at least for big installations).