

Open Remote Collaboration Tool OpenRCT

<http://www.openrct.org>

**Installation Manual
v1.6.0**

{amsler,walters}@cs.ucdavis.edu

University Of California, Davis

January 24, 2005

Table of Content

| | | |
|-------|--|----|
| 1 | Introduction..... | 3 |
| 2 | OpenRCT Client | 4 |
| 3 | OpenRCT Server..... | 5 |
| 4 | OpenRCT Database | 6 |
| 5 | OpenRCT Admin | 7 |
| 6 | Download Source Code | 8 |
| 7 | Server Installation..... | 9 |
| 7.1 | OmniORB and OmniNotification Installations | 9 |
| 7.2 | PostgrSQL Installation | 10 |
| 7.3 | Apache and PHP Installation | 12 |
| 7.4 | OpenRCT Server Installation | 13 |
| 7.4.1 | Running The Server | 14 |
| 8 | Client Installation | 16 |
| 8.1 | Running The Client | 16 |
| 9 | Client Demo..... | 18 |
| 10 | Index..... | 19 |

1 Introduction

The Open Remote Collaboration Tool (OpenRCT) is a multidisciplinary effort to enhance collaboration - between students working together, between students and instructional staff, and between researchers who are not co-located in time and space. OpenRCT is a platform-independent, multimedia tool that supports synchronous and/or asynchronous communication.

Users can engage in class structured team or group collaboration sessions utilizing the following tools:

- Multilingual chat
- Record and playback sound messages
- Multilingual text pad
- URL invocation
- File transfer
- Threaded Discussion Forums

2 OpenRCT Client

The OpenRCT client allows the user to login and to interact with the OpenRCT server. When the user starts the OpenRCT client, the authentication window appears. A valid “Login” and “Password” need to be entered in order to proceed with the login process. Once authenticated, the user is represented with the client’s main application window. This window provides three interactive sections called “User View”, “Class View”, and “Active Sessions”. In addition, it shows the “Broadcast Channel” through which vital server and client information are published.

The “User View” provides information about online/offline users, who are enrolled in the same classes as the user who is logged in.

The “Class View” lists all the classes in which the user is enrolled. In addition, it shows a per class level listing of groups and teams as well as their active/inactive users.

The “Active Sessions” shows all the interactive and structured collaboration processes and provides the means to access tools such as the chat module.

3 OpenRCT Server

The OpenRCT server facilitates the organizational structure that allows OpenRCT clients to efficiently collaborate. The server's middle ware, implemented in CORBA, is the access point for all clients. A PostgreSQL database is the server's backend, where it stores al the client interactions as well as the server's state.

4 OpenRCT Database

The OpenRCT database is based on the relational model, which consists of a collection of tables. The database stores the entire user-generated context as well as some OpenRCT client/server specific state information.

5 OpenRCT Admin

The OpenRCT admin tool provides an Internet based user and data administration tool. It has three different access levels, which allow user level specific administration capabilities. The three administration levers are:

- Admin
- Manager
- User

The admin level gives full access to all the OpenRCT data and state information that is stored in the database. The manager level gives access to information that is managed by the manager. The user level provides the means for a user to change his/her user profile and to access the user specific context.

6 Download Source Code

The source code is available via subversion. You can access the OpenRCT subversion repositories here:

<http://www.openrct.org/subversion.html>

In order to checkout the various modules, execute the following commands:

- openrct-client (trunk)
 - svn checkout <http://www.openrct.org/subversion.html>
- openrct-server (trunk)
 - svn checkout <http://www.openrct.org/subversion.html>
- openrct-admin (trunk)
 - svn checkout <http://www.openrct.org/subversion.html>
- openrct-db
 - svn checkout <http://www.openrct.org/subversion.html>

7 Server Installation

We suggest that you follow the following installation order of software packages that OpenRCT depends on.

- OmniORB
- OmniNotification
- PostgreSQL
- Apache with PostgreSQL and PHP support
- OpenRCT Server

7.1 OmniORB and OmniNotification Installations

First, we need to get the source code from the omniORB and omniNotify CVS repository. Please follow the instructions on this page:

http://omninoify.sourceforge.net/nfy_download.html

Now, we have to setup several environment variables. Add the following lines to your:

`$HOME/.cshrc`

```
setenv    PATH                ${PATH}:/opt/omni/bin
setenv    PATH                ${PATH}:/opt/omni/src/services/omniNotify/bin
setenv    LD_LIBRARY_PATH     /opt/omni/lib/
setenv    LD_LIBRARY_PATH     ${LD_LIBRARY_PATH}:/opt/omni/src/services/omniNotify/lib
setenv    SHLIB_PATH          /opt/omni/lib/
setenv    SHLIB_PATH          ${SHLIB_PATH}:/opt/omni/src/services/omniNotify/lib
setenv    OMNINAMES_LOGDIR    /opt/omniORB/tmp
setenv    OMNIEVENTS_LOGDIR   /opt/omniORB/tmp
setenv    OMNIORB_CONFIG      /opt/omniORB/omniORB.cfg
```

NOTE: The last three lines deal with omniORB configuration files and directories, which we have not setup yet.

So far, we downloaded all the necessary source code so that we can start to configure and to build omniORB and omniNotify.

```
cd /opt/omni
./configure
make
```

```
cd /opt/omni/src/services/omniNotify
./configure --with-omniorb=/opt/omni
make
```

Once you compiled omniORB, and omniNotify, the build and installation process is completed.

Now, we have to deal with a few omniORB configuration directories and one configuration file. First, we create the directories. Execute the following commands:

```
mkdir /opt/omniORB
mkdir /opt/omniORB/tmp
```

Then copy the sample configuration file “/opt/omni/sample.cfg” to “/opt/omniORB/omniORB.cfg”. Now, edit the “/opt/omniORB/omniORB.cfg” file. You have to add one entry in the “Client Side Options”. At the end of the “InitRef” section, you have to add the following entry:

```
InitRef = NameService=corbaname::192.168.1.10:2250
```

NOTE: You have to change the IP and Port number to match your setup. These are relevant to the Naming Service’s runtime setup.

Now, we have finished the installation and configuration process.

Before we run the RCT server, we need to start, initialize, the Naming Service once. Execute the following command:

```
omniNames -start 2250
```

Note: Make sure the port number, in the above case 2250, matches the port number in the omniORB.cfg file.

After you have executed the above command, you can kill omniNames.

7.2 PostgrSQL Installation

Download the latest PostgreSQL source code from:
www.postgresql.org

As of this writing the latest version is:
postgresql-7.3.3.tar.gz

Unpack the source code:
tar xvzf postgresql-7.3.3.tar.gz

Change to the postgresql-7.3.3 directory. Then you can configure the source code by executing the following command:
./configure --enable-multibyte

In order to build the source code execute the following command:
gmake

In order to install the database, you have to become root and then execute the following command:

```
gmake install
```

Now that you have installed the database, we need to do some configuration. First you need to add a user called “postgres”. Make sure that the user does not exist yet. If the user exists, you can skip this step. To add a new user, execute the following command:

```
adduser postgres
```

NOTE: You have to be root to add a user. Also make sure that the password is set for the user “postgres”.

Now that we have created a user called “postgres”, we can login as this user to finish up the database configuration. In order to login as the user “postgres”, execute the following command:

```
su – postgres
```

Now we can initialize the database. You are logged in as “postgres” now, then execute the following command to initialize the database.

```
/usr/local/pgsql/bin/initdb -E UNICODE -D /home/postgres/data
```

Now we can start the database by executing the following command:

```
/usr/local/pgsql/bin/postmaster -D /home/postgres/data&
```

Now we can create the a specific db by executing the following command:

```
/usr/local/pgsql/bin/createdb -E UNICODE rctdb
```

At this point all the necessary installations and configurations are done. What’s left is to load the actual db from a SQL file. You can get the RCT SQL file called rct.sql from CVS. Once you obtained the rct.sql file, you can load the table definitions in the created database. Execute the following command:

```
/usr/local/pgsql/bin/psql <dbname> < <SQL file>
```

eg.

```
/usr/local/pgsql/bin/psql rctdb < rct.sql
```

Now you have successfully installed, configured, and initialized the database. When you need to start the database in the future, just execute the following command:

```
/usr/local/pgsql/bin/postmaster -i -D /home/postgres/data&
```

NOTE: The “-i” parameter allows the web based admin tool to access the DB as well. Also make sure that you are logged in as the “postgres” user when you execute these commands.

Now you need to let the system know where PostgreSQL is installed. The easiest way is to add an entry to:

```
/etc/ld.so.conf
```

At the end of “/etc/ld.so.conf”, add the following:

/usr/local/pgsql/lib

7.3 Apache and PHP Installation

Download the latest Apache web server source code from:
<http://www.apache.org>

Download the latest PHP source code from:
<http://www.php.net/>

As of this writing the latest versions are:

Apache: 1.3.27
PHP: 4.3.2

Unpack the source code:
tar xvzf apache_1.3.27.tar.gz
tar xvzf php-4.3.2.tar.gz

Now that we have the source code, we can start with the configuration process. Change the directory to apache_1.3.27. Then execute the following command:
./configure

Now you have to change the directory to the PHP source code. Execute the following command:
cd ../php-4.3.2

Now we have to configure the PHP source code. Execute the following command:
./configure --with-pgsql --with-apache=../apache_1.3.27 --enable-track-vars --without-mysql

Then we have to compile the PHP source code. Execute the following command:
make

In order to install PHP, you have to become root and then execute the following command:
make install

Now, we have configured and installed PHP. Next step is to configure and to install Apache. Change the directory to the Apache web server source code. Execute the following command:
cd ../apache_1.3.27

Now we have to configure the Apache source code. Execute the following command:

```
./configure --activate-module=src/modules/php4/libphp4.a
```

Then we have to compile the Apache source code. Execute the following command:

```
make
```

In order to install the Apache web server, you have to become root and then execute the following command:

```
make install
```

The Apache server is now installed. We only have a couple more steps to go through, and then we have a working PostgreSQL, PHP, and Apache web server. We have to tell Apache how to deal with PHP scripts. We need to add a line to the httpd.conf file located in:

```
/usr/local/apache/conf/
```

You will see the following line commented out. In order to have Apache work with PHP4 files, you need to uncomment the line:

```
AddType application/x-httpd-php .php
```

We can check that PHP will be enabled and working with this command by executing the following command:

```
/usr/local/apache/bin/apachectl configtest
```

It should return with: "Syntax OK"

Now, we are ready to start the Apache server by executing the following command:

```
/usr/local/apache/bin/apachectl start
```

Now, the Apache web server should be running. You can check all the Apache web server configuration options in the file called httpd.conf. One thing you might want to change is the port number where the Apache server runs on.

7.4 OpenRCT Server Installation

Assuming you have downloaded the latest OpenRCT Server source code via Subversion, and have installed all the required software packages such as omniORB, omniNotify, and PostgreSQL, we can build the OpenRCT Server now. Change to the OpenRCT Server source code directory:

```
cd OpenRCTServer-X.Y.Z
```

Now you can configure the build process by executing the following command:

```
./autogen.sh
```

If the configure script finishes without any errors, we can compile the source code. Execute the following command:
make

This should build the OpenRCT server executable, rctd, in the “src” subdirectory.

Now you have to create the following directory structure:

```
mkdir /opt/rct  
mkdir /opt/rct/images  
mkdir /opt/rct/data
```

Make sure that these directories have permissions for the user that runs openrct to read and write too.

7.4.1 Running The Server

It is important to note that the OpenRCT server does not work from behind a firewall if clients outside the firewall try to access it. Specifically, if the PC has an assigned IP number from a private address space, the server will not function correctly. Here is a list of private address space IP numbers:

10.0.0.0 – 10.255.255.255

172.16.0.0 – 172.31.255.255

192.168.0.0 – 192.168.255.255

Once you have determined that the server’s PC is using an IP number that is not within any of the private address spaces, you are ready to start the server.

To run the server, change to the “src” subdirectory:
cd src

Then, you can start the OpenRCT Server by executing the following command:
./start-rct.sh

The “start-rct.sh” script starts three processes:
omniNames
notifyd

rctd

If you want to stop the OpenRCT Server, execute the following command:

```
./kill-rct.sh
```

This script stops all the above-mentioned processes.

8 Client Installation

NOTE: The following installation procedure has been tested on Linux, Windows, and Mac OS X

First you need to install Java on your system. The OpenRCT client requires the following Java version:

Java SDK v1.4.2_06

You can download it from:

<http://java.sun.com/j2se>

There, you will find specific versions for most of the operating systems. Make sure that you download the SKD and not just the JRE! If you want to run the OpenRCT Client on Mac OS X, you don't need to download Java since Mac OS X has it already installed. On a Mac OS X system, make sure that you upgrade the system to get the latest Java support.

Note: We recommend that you uninstall any previously installed Java SDK before installing the above-mentioned Java version. This only applies for Windows, UNIX, and Linux users.

At this point, you have downloaded and installed Java SDK. Next step is to download the OpenRCT Java Client. You can download the OpenRCT client from:

http://sourceforge.net/project/showfiles.php?group_id=56303

The client archive is available in both "ZIP" and "TAR.GZ" format. For Windows systems, download the client archive file with the ".zip" extension. For Linux, UNIX, and Mac OS X systems, download the client archive file with the ".tar.gz" extension. After you have downloaded the OpenRCT client, you have to unzip/untar the archive file

8.1 *Running The Client*

It is important to note that the OpenRCT client does not work from behind a firewall. Specifically, if the PC has an assigned IP number from a private address space, the client will not function correctly. Here is a list of private address space IP numbers:

10.0.0.0 – 10.255.255.255

172.16.0.0 – 172.31.255.255

192.168.0.0 – 192.168.255.255

When you are ready to start the client:

Windows: Double click on the “StartOpenRCT.bat” file.

Linux / UNIX: Execute the “StartOpenRCT.sh” file from within a terminal.

MAC OS X: Execute the “StartOpenRCT.sh” file from within a terminal.

9 Client Demo

We provide a test OpenRCT server setup where you can live demo the OpenRCT client. If you have not downloaded and installed the client yet, follow the instructions in this document to do so. Once you have downloaded and installed the client, follow the instruction in this document on how to run the client.

Once you have started the client, you are represented with the login window. You are prompted to enter a valid “Login” and “Password”. We provide a few user accounts on our test server that you can use. The following is a list of available logins and passwords:

| Login: | Password: |
|---------------|------------------|
| s1 | s1 |
| s2 | s2 |
| ... | ... |
| s10 | s10 |

When you start the client, it will connect to the preset server. For a complete list of available OpenRCT server, please check our web site at <http://www.openrct.org>.

10 Index

| | |
|--|----|
| Introduction | 3 |
| Client Demo | 18 |
| Client Installation..... | 16 |
| Running The Client | 16 |
| Download CVS..... | 8 |
| Introduction | 3 |
| OpenRCT Admin..... | 7 |
| OpenRCT Client..... | 4 |
| OpenRCT Database | 6 |
| OpenRCT Server | 5 |
| Server Installation | 9 |
| Apache and PHP Installation | 12 |
| OmniORB and OmniNotification Installations | 9 |
| OpenRCT Server Installation | 13 |
| Running The Server | 14 |
| PostgreSQL Installation | 10 |