

Reciprocal Net Site Software

release 0.9.1

User Guide

Eric Koperda and John Bollinger

March 2009

revised April 2009



THE NATIONAL SCIENCE DIGITAL LIBRARY





INTRODUCTION	5
Project details	6
System requirements	7
Software architecture.....	8
New features of interest in release 0.9.1	10
INSTALLATION	11
Overview	12
Upgrading from a previous version.....	12
Preparatory steps for new sites.....	12
Installing GNU/Linux from scratch	13
Customizing yum	16
Downloading GNU/Linux updates.....	18
Downloading additional software packages.....	19
Customizing Firefox and the Java plug-in	20
Downloading software updates (again).....	20
Customizing mail settings	21
Customizing MySQL settings	22
Customizing Tomcat settings	23
Customizing Httpd settings	24
Customizing Reciprocal Net site software	24
Initializing Reciprocal Net site software	25
Synchronizing with the Reciprocal Net site network.....	25
Conclusion.....	26
UPGRADING FROM A PREVIOUS VERSION.....	27
Overview of the upgrade process	28
Upgrading from legacy releases	28
Backing up Reciprocal Net data.....	28
Reconfiguring yum.....	29
Upgrading and reinstalling Reciprocal Net.....	31
Downloading GNU/Linux updates.....	31
Merging configuration files.....	32
Upgrading database schema	33
Synchronizing with the Reciprocal Net site network.....	34
Wrapping up.....	34
USING THE SITE SOFTWARE	35
Terminology	36
Initial administrative tasks	36
Creating a new sample	38
Importing metadata from CIF files.....	41
Attached data files.....	42
About chemical formulae	43
About space group symbols	44
About units of measure	45
ADMINISTERING THE SITE SOFTWARE	46
Web site customization	47
Daemon startup order	47

Ports and firewalls..... 47

Backups 48

Creating custom (locally-defined) metadata fields 49

Configuring sample number auto-generation..... 51

How your site interacts with the Site Network..... 51

About recipnet.sitegrant files 55

Configuration directives in recipnetd.conf 56

Configuration directives in web.xml 69

Format specification for .crt files 75

MISCELLANEOUS..... 79

 Technical support 80

 Development team 80

 Other contributors 80

 Acknowledgements 81

 License 82



Introduction

Reciprocal Net is a collection of molecular structure information that is affiliated with the National Science Digital Library. The *Reciprocal Net Site Network* is an online community that maintains a distributed database of crystallographic data. The *site software* described by this manual allows the operator to participate in the distributed database and is intended to be run at crystallographic facilities (that analyze crystals submitted by research chemists) located at major research universities.

The *Reciprocal Net site software* is a web application that can be used to track crystal samples submitted to your laboratory for analysis, and can be configured to allow the sample provider access to the data you collect. Many sample providers find the site software's web-based visualization features particularly useful. Chief among these features is JaMM, a Java applet that displays molecular models and can generate ray-traced 3-D images, ORTEP diagrams, and line drawings.

The Reciprocal Net Site Network is a *distributed* database, which means that it's easy for your laboratory to contribute the sample data it collects to the public. Crystal samples whose data you choose to let the public access will automatically become searchable via <http://www.reciprocalnet.org/>, the master Reciprocal Net server. Sample data that you do not explicitly release to the public does not leave your server and remains entirely under your control.

Project details

The Reciprocal Net project was started by a grant from the NSF in January 2002 and was integrated into the National Science Digital Library (see <http://www.nsdlib.org/>) in 2003. The Reciprocal Net project currently is based at Indiana University, although we hope the effort will evolve to a self-supporting, community-managed model.

The Reciprocal Net project, so far as the National Science Digital Library (NSDL) is concerned, comprises three *collections*:

- A collection of molecular structure data, where the structures solved and ready to be visualized. Submissions into this collection originate from respected crystallography facilities, generally affiliated with major universities.
- A collection of educational modules – sequences of web pages – that instruct students on specific aspects of chemistry. Submissions into this collection originate from professional educators in primary schools, secondary schools, and universities.
- A collection of visualization programs for rendering molecular structures. Submissions into this collection generally originate from computer programmers associated with the field of chemistry in some way.

The latter two collections are managed entirely by Reciprocalnet.org, the “portal,” and are outside the scope of this manual. The first collection, the one that contains molecular structures, is the focus of this document.



This Reciprocal Net *site software*, distributed as the *recipnet* package, is designed to be deployed at crystallography facilities and to integrate into their daily workflows. By using the Reciprocal Net site software for their day-to-day operations, these crystallography facilities have an easy means for contributing structures to Reciprocal Net's public collection. To date, roughly 250,000 molecular structures have been published in the scientific literature. How many more have been analyzed but never made public because they weren't sufficiently interesting? Reciprocal Net is an outlet for all structures, whether published or not. As a public educational resource, this comprehensive collection of molecular data will be unparalleled.

It is safe for your laboratory to use the Reciprocal Net site software even if it intends to make only a few structures public at first – the software was designed for this. No one, not even the Reciprocal Net Coordinator, can access structure data on your server to which you (the lab administrator) have not permitted it. You control who may access data on a per-structure basis: whether only fellow crystallographers in your lab, the chemist who submitted the crystal, or the general public. We recommend that labs institute a policy that structure data analyzed by them becomes publicly visible via Reciprocal Net when either a) the structure is published in a scholarly journal, or b) after five years. In this way, intellectual property rights of the submitting chemist can be preserved while the usefulness of Reciprocal Net as an educational resource continues to grow. (Of course, the five-year rule is only a suggestion; the software will not make your structure data visible to the public until you instruct it to do so.)

System requirements

The Reciprocal Net site software is a web application written primarily in Java, designed to be installed on an existing web server running the GNU/Linux operating system. It is distributed in Red Hat Package Manager (RPM) format. An online Yellow Dog Updater Modified (YUM) repository is available.

Sever hardware requirements

- System
 - minimum: any modern Intel-based desktop PC.
 - recommended: server-class machine with an uninterruptable power supply for high availability.
- Memory
 - recommended: at least 512 MB RAM
- Disk
 - minimum: at least 5 GB free hard drive space.
 - recommended: redundant disk array for high availability
- Processor
 - minimum: Pentium 4-class 32-bit capable processor
 - optional: 64-bit-capable processor (e.g. EM64T, AMD64). However, this release of Reciprocal Net site software does not take advantage of 64-bit-capable processors.

Sites that are particularly concerned about availability and reliability are advised to select server-class hardware with redundant disk arrays.

Server operating system requirements

Several distributions of the GNU/Linux operating system have been tested extensively with this release of the Reciprocal Net site software and are known to be compatible:

- Red Hat Enterprise Linux 5, i386 edition (including release 5.3)
- CentOS 5, i386 edition (including release 5.2)
- Scientific Linux 5, i386 edition (including release 5.3)
- Fedora Core 10, i386 edition

Use of other GNU/Linux distributions and even other operating systems may be possible but cannot be supported by the Reciprocal Net project. Note that only the **i386** editions of GNU/Linux operating systems are supported at this time. These are able to run on both 32-bit-capable and 64-bit-capable Intel-compatible processors. Although some sites have reported success running Reciprocal Net sites on **x86_64** editions of the GNU/Linux operating system, such configurations have not been tested and thus are not recommended.

Older GNU/Linux distributions are no longer supported by the Reciprocal Net project. In particular, use of Red Hat Linux 8 is discouraged because its product support ceased in year 2003, and use of Fedora Core 1 is discouraged because its product support ceased in year 2005. Existing Reciprocal Net sites that may be running these obsolete operating systems are urged to upgrade.

Server support software requirements

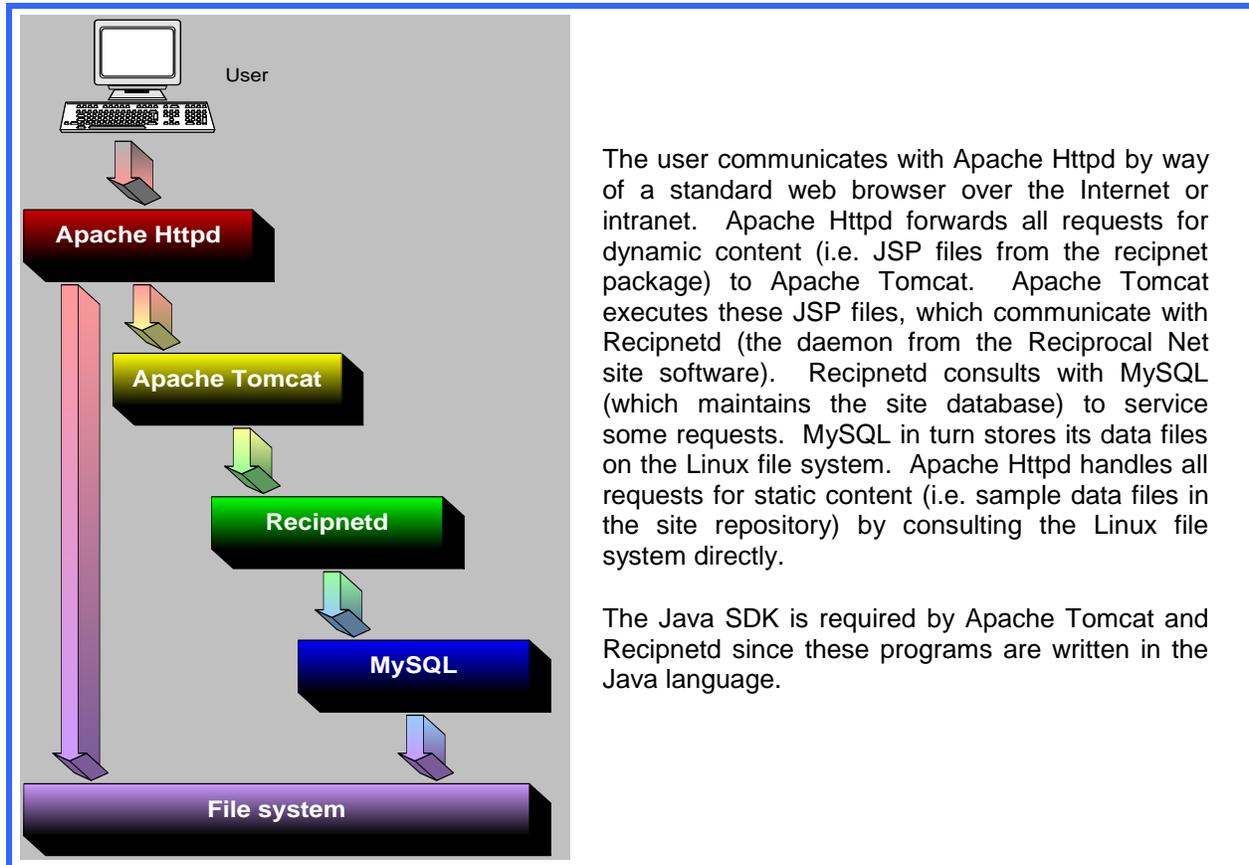
- A web server daemon. *Apache Httpd 2.2* from the Apache Software Foundation has been tested extensively and is supported.
- A servlet container daemon or J2EE daemon. *Apache Tomcat 5.5* from the Apache Software Foundation has been tested extensively and is supported. Note that some software packages are capable of fulfilling both the web server requirement and the servlet container requirement.
- A Java virtual machine (VM) compatible with J2SE specification 1.6.0 or higher. *JDK 6* from Sun Microsystems has been tested extensively and is supported.
- A SQL database server for which a Java (JDBC) driver is available. *MySQL 5.0* from MySQL AB has been tested extensively and is supported.
- The most recent release of the *Reciprocal Net site software*.

Software architecture

Most features of Reciprocal Net site software are designed to be accessed across the Internet from a standard web browser. Some molecular visualization features require special support on the client side for Java applets, and unfortunately not all platforms are compatible. The most current client-side compatibility list is maintained online at <http://www.reciprocalnet.org/networkinfo/docs/> as *Technical Bulletin #02*. Briefly, full support is available for web browsers running under recent versions of Microsoft Windows, Red Hat Linux, and SGI IRIX, and limited support is available under Apple MacOS 9 and MacOS X.

As a modern web application, Reciprocal Net site software interoperates with a number of other software applications on the server side. Some examples include Java compilers and runtime environments, Java libraries, J2EE containers, web-serving engines, and SQL database engines. The system administrator must be familiar with all the applications on the server in order to achieve a stable and balanced configuration.

The following diagram illustrates how the site software (*recipnetd*) integrates with other software packages a typical site server.



The user communicates with Apache Httpd by way of a standard web browser over the Internet or intranet. Apache Httpd forwards all requests for dynamic content (i.e. JSP files from the recipnet package) to Apache Tomcat. Apache Tomcat executes these JSP files, which communicate with Recipnetd (the daemon from the Reciprocal Net site software). Recipnetd consults with MySQL (which maintains the site database) to service some requests. MySQL in turn stores its data files on the Linux file system. Apache Httpd handles all requests for static content (i.e. sample data files in the site repository) by consulting the Linux file system directly.

The Java SDK is required by Apache Tomcat and Recipnetd since these programs are written in the Java language.

New features of interest in release 0.9.1

This is a minor maintenance release for Reciprocal Net site software. It contains limited new functionality. Significant changes include:

- Increased reliability during inter-site replication.
- New troubleshooting tools useful when inter-site replication fails.
- Enhanced support for lab migrations and site deactivations in the Site Network.
- Eliminated an unsustainable resource barrier involving site grant files for new sites.
- Improved dependency interactions with upstream software vendors to reduce installation effort.
- Fixed a bug that caused 100% CPU utilization following unclean shutdowns by apache *httpd*.
- Fixed a bug related to file names during file uploads by authenticated lab users.
- Fixed a bug that prevented customization of configuration directives *SitDbUsername*, *SamDbUsername*, and *RepDbUsername*.
- Compatibility with the newest distributions of GNU/Linux, including Red Hat Enterprise Linux 5.3, Fedora Core 10, and the newest supporting software packages.



Installation

Overview

The instructions in this chapter will direct the reader through a complete installation beginning with an empty hard disk drive. It will address installation and configuration of an operating system (Red Hat Enterprise Linux 5.3 Server i386), Reciprocal Net site software, and the recommended supporting packages (Apache Httpd, Apache Tomcat, MySQL, Sun's Java SDK, etc.). The instructions describe one way to assemble a Reciprocal Net site but there are many more.

Substitution of a “clone” distribution of Red Hat Enterprise Linux, such as CentOS or Scientific Linux, is straightforward. In most cases only the on-screen textual labels have changed. These instructions will address the few key distinctions as they arise.

Experienced system administrators may opt to deviate from these instructions more dramatically still to suit local preferences, but should bear in mind that such deviations may frustrate the Reciprocal Net project's efforts to assist them.

Upgrading from a previous version

If a previous release of the Reciprocal Net site software already is installed on your server, the upgrade directions are considerably different from what's described in this chapter. Please skip ahead to the chapter titled *Upgrading from a previous version* on page 27.

Unlike some previous releases, with this release 0.9.1 there is no requirement to install a new operating system on your server.

Preparatory steps for new sites

If you have not done so already, you should consider consulting with your organization's information technology support group about this new server. They may have certain configuration preferences or tips to share with you, and they may be able to provide technical assistance with system administration to you. GNU/Linux system administration is not for the faint of heart.

You cannot bring a new Reciprocal Net site online until your crystallography laboratory has received permission to join the Reciprocal Net Site Network and has obtained a *site grant* file. This is a binary configuration file that is *not* downloadable online – you must request it separately. See page 55 for a more thorough discussion of what this file is and why you need it.

Only the Reciprocal Net Coordinator can issue site grant files. Currently the Coordinator function is being performed by the Molecular Structure Center at Indiana University. To obtain the file, send an e-mail to Dr. Maren Pink (info@reciprocalnet.org) telling her that you wish to join the Reciprocal Net site network. She'll contact you offline (probably via telephone or videoconference) just to say hello and to arrange for the file exchange. The



process normally takes a couple of business days. You will need to have received your *recipnet.sitegrant* file in order to complete the instructions in this chapter.

Installing GNU/Linux from scratch

Obtaining installation media

These instructions assume that you will install Red Hat Enterprise Linux 5.3, i386 Server edition from its five distribution CD-ROM discs. Red Hat Enterprise Linux is a commercial product; there are no free mirrors available on the Internet. If your server is owned by the Reciprocal Net project and bears an Indiana University inventory control tag (even though it may be loaned indefinitely to your institution), please contact Reciprocal Net technical support to receive a copy of Red Hat Enterprise Linux. Otherwise, please consult with your own organization's network administrators or software distribution center to obtain copies of the installation media.

If you do not have access to Red Hat Enterprise Linux, do not need business-class technical support, and are interested in a free alternative, there are a number of “clone” distributions available. Two examples are CentOS 5.3 and Scientific Linux 5.3, both of which have been tested and are fully compatible with Reciprocal Net site software. For these distributions, you can create the installation discs yourself by locating a mirror on the Internet, downloading *.iso* images for the five or six discs, and “burning” the images to CD-R's or DVD-R's using a CD-writing or DVD-writing program on your desktop computer.

If you do not have convenient access to an optical drive, please contact Reciprocal Net technical support for assistance.

Installing from DVD

Insert the installation DVD into your computer, power it on, and let the computer boot from the DVD-ROM drive. When you see the splash screen, press the **Enter** key to begin an installation. The system will trundle for a minute and then ask you to test your media. Highlight the **OK** button and press the **Enter** key to continue. In the next window, highlight the **Test** button and press the **Enter** key. The media test takes a few minutes to run. If you are installing from multiple CD's, repeat this process for each installation disc in the set. Then highlight the **Continue** button, and press the **Enter** key.

Next, the installation engine called Anaconda will load. It should detect your video adapter and attached monitor and then display a graphical installation wizard. On the welcome screen that appears, read the message displayed in the center, read the **Release Notes**, and then click the **Next** button.

On the language screen that appears, choose **English** in the left pane and click the **Next** button.

On the keyboard screen that appears, choose **U.S. English** in the left pane and click the **Next** button.

On Red Hat Enterprise Linux, an *Installation number* dialog box may appear. If so, consult with your organization's network administrator regarding software licensing arrangements. Either type the number he provides or skip ahead without entering a number.

An upgrade screen may appear if the hard disk drive previously had another GNU/Linux operating system installed. If this screen should appear, choose the *Install Red Hat Enterprise Linux Server* option and then click the *Next* button. Reciprocal Net recommends against an in-place upgrade of operating systems because fresh installations tend to be more stable and reliable, particularly if you backed up your Reciprocal Net site's data as described in the preceding section.

On the partitioning screen that appears, choose the *Remove all partitions on selected drives and create default layout* option and click the *Next* button. In the *Warning* dialog box that appears, click the *Yes* button. Experienced system administrators may prefer to manually partition and format their drives instead. If you choose the latter route, Reciprocal Net recommends that you place the */var* subtree on its own large partition. Molecular structure entries you create within the Reciprocal Net site software will be stored in the directory */var/recipnet/data*, and this repository can grow to be many gigabytes in size, particularly at more active crystallography labs that archive their raw data. At Indiana University's Molecular Structure Center, */var* partitions are never smaller than 50 gigabytes. You might find a swap partition useful also; at IU swap partitions are generally about 1 gigabyte in size.

In the network window that appears, configure the settings to match those of your local network. Consult with your organization's network administrator as needed. Note that use of DHCP is discouraged because Reciprocal Net site software acts as a web server and thus requires an IP address that does not change frequently. Reciprocal Net recommends that member servers have their IP address either configured statically or assigned a "sticky" address by a DHCP server. Note also that IPv6 is not utilized by Reciprocal Net servers and may be disabled to simplify configuration. Click the *Next* button when your network configuration is complete.

On the time zone screen that appears, select a time zone according to your local preferences, then click the *Next* button.

On the root password screen that appears, invent a root password for yourself. The *root* user account is used for administration of your operating system and has full access to the computer. You should choose a password that includes both letters and numbers, is hard to guess, and preferably is unique to this particular computer. Type your chosen password twice, once into each box, and click the *Next* button.

On the package selection screen that appears, you have the opportunity to specify which programs are to be installed on your server. Experienced system administrators should feel free to select or unselect any packages according to their local customs; any Reciprocal Net site software dependencies will be satisfied later in this chapter. On Red Hat Enterprise Linux, the default configuration with all options disabled is fine. On other distributions, clearing the check box for the *Office/Productivity* package group (if checked) may be desirable. Click the *Next* button when your package configuration is complete.



The computer will trundle for a moment and then the ready screen appears. Verify that you're ready to proceed and then click the *Next* button. When the *Required Install Media* window appears, read the message, verify that you have all needed discs ready, and click the *Continue* button.

A progress bar will appear and the computer will trundle for several minutes as the hard disk drive is formatted and packages are installed from disc. If there are several installation discs, insert each in turn as prompted.

The next screen that appears will inform you that installation is complete. Remove your installation disc from the computer's drive and click the *Reboot* button. Your computer will reboot.

First-boot configuration

As the computer boots, the *Welcome* wizard will appear. Click the *Forward* button.

On Red Hat Enterprise Linux, a *License Agreement* screen may appear. Read the text, select the *Yes, I agree to the License Agreement* option, and click the *Forward* button.

On the *Firewall* screen that appears, choose the *Enabled* option. Scroll down to select the *WWW (HTTP)* option in the box beneath to allow your computer to act as a web server. (You may need to scroll down to see all options.) Many system administrators find it convenient to select also the *SSH* option because this allows the computer to act as a Secure Shell server. SSH allows you to access a command line remotely and facilitates remote system administration. There is no need to select the *Secure WWW (HTTPS)* option. Click the *Forward* button. In the confirmation window that appears, click the *Yes* button.

On the *SELinux* screen that appears, choose the *Enforcing* option. SELinux is a security enhancement technology that imposes mandatory access controls on select daemons. Its use in *Enforcing* mode is recommended. Experienced system administrators who employ other security mechanisms may choose *Permissive* or *Disabled* as necessary. Click the *Forward* button. If a confirmation dialog box appears, click the *Yes* button.

On Red Hat Enterprise Linux, a *Kdump* screen may appear. By default the *Enable* checkbox is not selected. This is acceptable. Click the *Forward* button.

On the *Date and Time* screen that appears, set the clock appropriately. Then click the *Network Time Protocol* tab first. Select the *Enable Network Time Protocol* option and enter the name of your organization's NTP servers below. If your organization does not operate its own network time server, you are welcome to use the one operated by Indiana University: click the *Add* button and then type:

```
ntp.indiana.edu
```

. Click the *Forward* button when your date configuration is complete.

On Red Hat Enterprise Linux, a *Set Up Software Updates* screen may appear. If it does, follow the instructions of your network's administrator. Exact choices will depend upon your organization's network and software license arrangements. Click the *Forward* button. If a *Finish Updates Setup* screen appears, click the *Forward* button.

On the *Create User* screen that appears, you should set up a personal user account that you will use for routine activities other than system administration. Some organizations utilize network-based authentication rather than local authentication; for these details, consult with your network administrator. Once user accounts have been configured, click the *Forward* button.

A *Sound Card* screen may appear. If it does, the default settings frequently are acceptable. Click the *Forward* button, the *Finish* button, or the *Reboot* button, whichever is present.

On Red Hat Enterprise Linux, an *Additional CDs* screen may appear. No action on it is necessary. Click the *Finish* button or the *Reboot* button, whichever is present.

Some systems may require a reboot at this point. If a confirmation window appears, click the *OK* button and allow the computer to reboot.

When a login prompt appears, log in as the *root* user using the password specified earlier. You should see a GNOME desktop.

Depending on the details of your organization's software license for Red Hat Enterprise Linux, there may be a "bootstrap" script that needs to be run on the computer. Run the bootstrap script now. Consult with your organization's network administrator for the details.

Then continue on to the next section.

Customizing yum

Red Hat Enterprise Linux includes a tool called *yum* that automates software installations and routine patches. Yellow Dog Updater Modified, or *yum*, is a program that manages packages, resolves dependencies, and downloads new packages from the Internet as needed.

The *yum* program must be instructed how to download Reciprocal Net software. The Reciprocal Net project maintains a repository of RPM packages on its Internet web site. To tell *yum* how to access this repository, it is necessary to install two small *.repo* files on your system. From the GNOME desktop, launch the program *Applications/Accessories/Terminal* to get to a command prompt. At the command prompt, type the following command (all on one line):

```
wget -P /etc/yum.repos.d/ www.reciprocalnet.org/download
/ recipnet.repo
```

. Then, type the following command (all on one line):

```
wget -P /etc/yum.repos.d/ www.reciprocalnet.org/download
/ recipnet-jpackage-rhel5.repo
```

. **yum** now is able to access packages in the Reciprocal Net project's repository and also those in the JPackage project's repository.

The **yum** tool works best in conjunction with a plug-in called **yum-priorities**. From the GNOME desktop, navigate to the **Applications** menu and launch the **Add/Remove Software** program. In the window that opens, click the **List** tab. The program will trundle for a moment and then display a list of all software packages available for download. Scroll through the list; find and select **yum-priorities**. Click the **Apply** button. In the **Package selections** window that appears, click the **Continue** button. A **Downloading packages** window will appear with a progress bar.

An **Import key** window may appear asking permission to trust software published by the Reciprocal Net project. Verify that the key ID reads **0x4889CA78**. Click the **Import key** button. Finally a completion message appears. Click the **OK** button. Close the **Add/Remove Software** program.

Next, on Red Hat Enterprise Linux, from the GNOME desktop, navigate to the **Applications/Accessories** menu and launch the **Text Editor** program. In the window that opens, click the **Open** button. Navigate to the **/etc/yum/pluginconf.d/** directory and open the file named **rhnpugin.conf**.

Edit the contents of the **rhnpugin.conf** file to enable repository protection. Scroll to the bottom of the file, in the section labeled **[rhel-i386-server-5]**, and add the following text on a new line:

```
priority = 1
```

. Then click the **Save** button.

TIP → On some GNU/Linux distributions, there may be no file named **pluginconf.d**. Instead, check the directory named **/etc/yum.repos.d/** for repositories associated with your operating system distribution. Open those repository configuration files and add **priority** directives (as above) to each repository declaration.

Next, in the **Text Editor** program window, click the **Open** button. Navigate to the **/etc/yum/plugins.d/** directory and open the file named **priorities.conf**.

Edit the contents of the *priorities.conf* file to enable enhanced obsoletes checking. Scroll to the bottom of the file, in the section labeled *[main]*, and add the following text on a new line:

```
check_obsoletes = 1
```

. Then click the *Save* button.

Most system administrators prefer to not be bothered with watching for security updates and staying on top of patches. It is easy to configure *yum* to update your server automatically, in the background.

In the *Text Editor* program window, click the *Open* button. Navigate to the */etc/yum/* directory and open the file named *yum-updatesd.conf*.

TIP→ On some GNU/Linux distributions, there may be no file named *yum-updatesd.conf*. You can fix this problem by installing another software package. Using the *Add/Remove Software* program as before, download and install the package named *yum-updatesd*. Then continue with these instructions.

Edit the contents of the *yum-updatesd.conf* file to enable automatic updates. Find the line that begins with *do_update* and modify this line to read:

```
do_update = yes
```

. Find the line that begins with *do_download* and modify this line to read:

```
do_download = yes
```

. Find the line that begins with *do_download_deps* and modify this line to read:

```
do_download_deps = yes
```

. Finally, click the *Save* button and close the *Text Editor* program.

Downloading GNU/Linux updates

The responsible system administrator applies operating system patches to his system as they are released in order to protect his system against crackers who might attempt to exploit newly-discovered vulnerabilities in the software his server runs. Timely patch management is a crucial component of contemporary system administration practice. As of the time of this writing, Red Hat Enterprise Linux 5 has dozens of known security vulnerabilities that can be fixed by downloading patches.



To update your computer's software packages, from the GNOME desktop, find the *Applications* menu in the top-left corner of the screen. Navigate to the *Applications/System Tools* menu and launch the *Software Updater* program.

The computer will trundle for a moment and then display a list of updates available. Verify that all of the available packages are selected. Click the *Apply updates* button. A progress bar appears and packages are downloaded.

If it did not appear earlier, an *Import key* window may appear asking permission to trust software published by the Reciprocal Net project. Verify that the key ID reads *0x4889CA78* . Click the *Import key* button.

A second *Import key* window may appear for the JPackage project. Verify that the key ID reads *0xC431416D* . Click the *Import key* button.

On some distributions of GNU/Linux, a third *Import key* window may appear for the operating system distribution. Verify that the key matches the one used by your distributor. Click the *Import key* button.

If a *Reboot recommended* window appears, click the *Reboot now* button. Wait while the computer reboots. Repeat these steps – rebooting and running the *Software Updater* – until there are no more updates to be installed.

Downloading additional software packages

Reciprocal Net site software is distributed as a set of three RPM packages. Those RPM packages depend upon several dozen other RPM packages that are published and distributed by other organizations in the open-source world. The easiest way to obtain precisely those packages your server needs is to download them using *yum*.

Next, from the GNOME desktop navigate to the *Applications* menu and launch the *Add/Remove Software* program. In the window that opens, click the *List* tab. The program will trundle for a moment and then display a list of all software packages available for download.

Scroll through the list and select the following three packages: *recipnet-site-server*, *recipnet-site-utils*, and *recipnet-site-webapp*. Click the *Apply* button. In the confirmation window that appears, click the *Continue* button. A *Resolving dependencies* window will appear with a progress bar; this step may take a moment.

In the *Dependencies added* window that appears, note that a few dozen other packages have been detected as necessary for your system and will be installed at the same time. Click the *Continue* button. A *Downloading packages* window will appear with a progress bar; this step may take several minutes depending upon the speed of your Internet connection.

If it did not appear earlier, an *Import key* window may appear asking permission to trust software published by the JPackage project. Verify that the key ID reads *0xC431416D* . Click the *Import key* button.

If it did not appear earlier, a second **Import key** window may appear for the Reciprocal Net project. Verify that the key ID reads `0x4889CA78` . Click the **Import key** button.

An **Updating software** window will appear with another progress bar. This step also may take a few minutes. Finally, a **Software installation completed** window will appear. Click the **OK** button and quit the **Package Manager** program.

Customizing Firefox and the Java plug-in

On Red Hat Enterprise Linux, the default graphical web browser is Mozilla's Firefox. This browser will not display web sites that utilize Java plug-ins without further customization. Because the Reciprocal Net web application does make use of Java plug-ins, you should consider whether you ever will access the web application from the server console, or whether all access will occur via other desktop computers. Continue with this section only if you choose the former.

First, verify that the Java plug-in is installed. Launch the **Add/Remove Software** program as before and look on the **List** tab for a package named **java-1.6.0-sun-plugin** . If it is not already present on your system, download and install this package now.

TIP→ On some GNU/Linux distributions, there may be several different editions present of the Java Runtime Engine (JRE). Not all JRE's have the same capabilities or directory structure. You can control which JRE is the default one by running: **alternatives --config java** . Reciprocal Net site software is designed to work with the JRE published by Sun Microsystems and repackaged by the JPackage Project.

Then, go to a command prompt and type the following command (all on one line):

```
ln -s /usr/lib/jvm/jre/plugin/i386/ns7/libjavaplugin_oji.so  
/usr/lib/firefox-3.0.7/modules/
```

. If your system's version of Firefox is different than 3.0.7, modify the final argument accordingly.

Now Java applets embedded within web pages should display correctly. Continue to the next section.

Downloading software updates (again)

In the preceding sections, a number of new software packages were downloaded and installed onto your server. Now it is necessary to verify that your system is equipped with the most recent releases of those software packages.



As before, navigate to the *Applications/System Tools* menu and launch the *Software Updater* program. Verify that all of the available packages are selected. Click the *Apply updates* button. A progress bar appears and packages are downloaded.

TIP→ Although Red Hat Enterprise Linux is generally compatible with the packages published by the JPackage Project, the same is not true for all GNU/Linux distributions. On some distributions, cryptic but persistent package dependency errors may appear. The short-term solution is to unselect available updates one-by-one until you discover some combination of packages that can install successfully onto your system. Longer-term, such disagreements tend to be resolved by operating system distributors and package maintainers.

After each round of updates, repeat the process until no further updates remain.

Customizing mail settings

Electronic mail addressed to the *root* user on your server should be directed to a “real” e-mail account read on a regular basis by the system’s administrator. Even if the server has been configured not to accept mail from external hosts (recommended), important notices from locally-running daemons still may be generated.

Continue with this section if you would find it convenient to have *root*’s mail forwarded to another e-mail account.

Launch the *Text Editor* program again. In the window that opens, click the *Open* button and then open the file named */etc/aliases* . Scroll to the bottom of the file and find the line that begins with the word *#root* . (The leading *#* symbol indicates that the line has been commented out.) Remove the comment symbol and change the line to read something like:

```
root: yourname@your.university.edu
```

. Click the *Save* button on the toolbar and then close the program.

Now it is necessary to activate the change you just specified. From the GNOME desktop, launch the program *Applications/Accessories/Terminal* to get to a command prompt. At the command prompt, type:

```
newaliases
```

. One line of output will appear and mail forwarding will take effect immediately.

Customizing MySQL settings

A default Reciprocal Net site software installation uses MySQL as its database engine. While it is possible to employ most any Java-accessible SQL engine with the site software, most administrators find it convenient to use the Red Hat-sanctioned version of MySQL. The MySQL packages should have been downloaded and installed earlier, if they weren't already present on your system.

The first step is to start the *mysqld* daemon. From the GNOME desktop, navigate to the *System/Administration* menu in the upper-left corner of the screen and launch the *Services* program. In the *Service Configuration* window that appears, look in the left-lower pane and find the *mysqld* item. Highlight *mysqld* and click the *Start* button on the program's toolbar. Click the *OK* button on the success message that appears a few seconds later. Set the *Service Configuration* program aside but leave it open.

The next step is to secure MySQL by setting the *root* passwords. Invent a *database root password* for yourself. This might be the same as GNU/Linux's *root* password, or it might be different (recommended). Go to a command prompt and type the following command:

```
mysqladmin -u root password 'your password'
```

, replacing *your password* with your desired password.

Some systems may be equipped with a second *root* account. Set the password on the second account by typing the following command:

```
mysqladmin -u root -h $HOSTNAME password 'your password'
```

. This second command may fail with an *Access denied* error message if there was no second *root* account; do not be alarmed. Continue forth.

Now, you almost certainly want to modify the way in which MySQL authenticates users because the default configuration can interfere with password-based authentication. At a command prompt, type:

```
mysql -u root -p
```

to open MySQL's interactive console. When prompted, type the database root password that you invented a moment ago. You should see a welcome message and a *mysql>* prompt. Type the following sequence of four commands:

```
USE mysql;

DELETE FROM user WHERE user='' OR host='localhost' OR
host='127.0.0.1';

UPDATE user SET host='%';

exit
```

. The last command should exit the interactive console and return to a shell command prompt.

Lastly, it is necessary to configure the MySQL daemon to auto-start every time the computer boots. At a command prompt, type:

```
chkconfig mysqld on
```

Customizing Tomcat settings

Apache Tomcat is a Java servlet container that enables Java applications to interact with web servers. Compared to previous versions of Tomcat, version 5.5 requires relatively little configuration.

Launch the Tomcat daemon by opening the *Service Configuration* program, highlighting *tomcat5*, and clicking *Start*.

TIP→ On some GNU/Linux distributions, as Tomcat starts it may report errors involving absent Java extensions. These are fixed by downloading and installing additional software packages through the *Add/Remove Software* program as before. For example, errors involving the *commons-el* Java extension are solved by installing the package named *jakarta-commons-el* and errors involving the *eclipse-ecj* Java extension are solved by installing the package named *eclipse-ecj*.

Next, ensure that the Tomcat daemon will auto-start every time the computer boots. At a command prompt, type:

```
chkconfig tomcat5 on
```

Customizing Httpd settings

Apache Httpd is a web server engine that answers requests from web browsers. Its web content lives beneath the `/var/www/html/` directory of the file system. You are free to place any web content you like on your server (so long as it doesn't interfere with the *recipnet* directory). In case your server has no other content, Reciprocal Net has installed a simple *index.html* file in that directory that redirects visitors to the site software web application. Thus, web content customization is optional but not necessary.

Start the Httpd daemon by launching the *Service Configuration* program, highlighting *httpd*, and clicking *Start*.

Next, ensure that the Httpd daemon will auto-start every time the computer boots. At a command prompt, type:

```
chkconfig httpd on
```

The resourceful system administrator will make many changes to the configuration of Apache Httpd beyond those described here – increasing performance and improving security are two common tasks. Plenty of excellent documentation about administering Httpd is available online at <http://httpd.apache.org/>.

Customizing Reciprocal Net site software

Locate the *recipnet.sitegrant* file you obtained earlier from the Reciprocal Net Coordinator and copy it to the directory named `/etc/recipnet/` on your server. This file contains your site's private encryption key and needs to be protected from prying eyes.

From the GNOME desktop, double-click on the *Computer* icon, then double-click on the *Filesystem* icon to access the file system. Navigate to the `/etc/recipnet/` folder on your computer. Find the file named *recipnet.sitegrant*, right-click it, and choose *Properties* from the popup menu. In the window that opens, click the *Permissions* tab. In the *Owner* section, verify that the owner is *root* and the corresponding access is *Read and write*. In the *Group* section, select *recipnet* and set the corresponding access to *Read-only*. In the *Others* section, set the access to *None*. Click the *Close* button.

There are six configuration files in the `/etc/recipnet` directory that control how different components of the Reciprocal Net site software operate. Many system administrators choose to modify these configuration files in order to optimize performance, customize logging, or add a visual brand to their site. At minimum, you should peruse the file named *recipnetd.conf* to verify that the default settings for the *recipnetd* daemon are appropriate for your system. See page 56 for a detailed reference on *recipnetd.conf*. Or, there is a place to display your organization's logo – see page 47 for more about the file named *sitesponsor.gif*.

Initializing Reciprocal Net site software

Go to a command prompt and type:

```
drecipnet init
```

. A window will appear with some explanatory text about the installation procedure. Read carefully through the message and answer *y* when it asks if you'd like to continue.

You will be prompted for the username and password of a user account that will be created for you. This user account will exist within your site's Reciprocal Net installation and will have *site administrator* privileges. (Reciprocal Net user accounts are unrelated to MySQL database user accounts, GNU/Linux user accounts, Tomcat user accounts, or Apache user accounts.) Many users choose to use their personal username (8 characters maximum) and password, although it's recommended that you choose a Reciprocal Net password that's different from your GNU/Linux account password. You can use this user account later to create additional Reciprocal Net user accounts so that your colleagues can access the system.

Next you'll be prompted to enter the *database root password* that you set earlier in this chapter. The program will trundle for a few seconds and generate many lines of output. If it doesn't end with the word **Done** then you have problems – read carefully through the output and try to find the error message. If no error is readily apparent on your console, then it's possible the error message was written to syslog instead, in which case the message could be found at the bottom of the */var/log/messages* file (in a default Red Hat Enterprise Linux installation).

Notice that while the init mode was running, it created three database user accounts for you and wrote the passwords to */etc/recipnet/recipnetd.conf* for safekeeping.

Synchronizing with the Reciprocal Net site network

The next step is to synchronize your server with the other servers in the Reciprocal Net Site Network. During synchronization, your server sends and receives messages over the Internet, processes any updates sent by the other servers, and tracks the samples publicized by the other servers.

The synchronization process may be lengthy because it depends upon the size of the Site Network and the activity of its member sites. As a benchmark, consider that in March 2009 a new site should expect to receive and process about 25,000 update messages. At a typical processing rate of 750 messages per minute, the expected processing time is 33 minutes. This number will only increase as the Reciprocal Net Site Network continues to grow.

Go to a command prompt and type:

```
drecipnet sync
```

. A bevy of status messages will be displayed, and any of them may seem to stall for several minutes. You'll know the process has concluded successfully when the word **Done** finally appears.

Conclusion

Finally it's time to start the Reciprocal Net daemon. Open the *Service Configuration* program and find **recipnetd** in the lower-left pane. Highlight **recipnetd** and click the **Start** button on the program's toolbar. In the confirmation window that appears a few seconds later, click **Ok**.

Congratulations! The Reciprocal Net site software is now running on your server. You can access it by pointing your web browser to <http://www.yourserver.edu/recipnet/>. You should be able to log in with the *site administrator* credentials you specified earlier in this chapter.

Most labs find it convenient to tuck their Reciprocal Net server away in a closet somewhere and interact with it exclusively via web browsers running on their desktop computers.



Upgrading from a previous version

Overview of the upgrade process

The instructions in this chapter assume your system runs Reciprocal Net site software 0.9.0 and you are upgrading to release 0.9.1. They also will assume your operating system is Red Hat Enterprise Linux 5.3 Server i386 and includes the typical supporting software packages. Substitution of a “clone” distribution of Red Hat Enterprise Linux, such as CentOS or Scientific Linux, is straightforward.

Because this release of Reciprocal Net site software is a minor maintenance release, there is no need to switch operating systems or reformat hard drives in order to upgrade. Release 0.9.1 is fully compatible with the same operating systems as was release 0.9.0. Most of the work is accomplished in-place by *yum*, the Yellow Dog Updater Modified program component of the GNU/Linux operating system.

It is necessary to stop several key system services for the duration of the upgrade. This will result in a service outage of your Reciprocal Net site, and it will appear to web surfers that your site is down. If many people rely upon your site, it may be prudent to schedule and announce downtime for the upgrade several days in advance. Scheduled downtime of two hours should be more than sufficient.

Upgrading from legacy releases

In case your system presently runs an older release of Reciprocal Net site software, such as 0.6.2, the upgrade procedure is more complicated. One solution is to upgrade from 0.6.2 to 0.9.0 in one step, according to the directions in the *0.9.0 User Guide*, then subsequently upgrade from 0.9.0 to 0.9.1 in a second step, according to this chapter.

Alternately, experienced system administrators may prefer to skip release 0.9.0 entirely and upgrade directly from 0.6.2 to 0.9.1. The Reciprocal Net RPM's will not complain. However, you will need to pay special attention to the operating system and supporting software packages. The following outline may be helpful:

- Backup the Reciprocal Net data on your existing site per the *0.9.0 User Guide*.
- Install a modern operating system and configure it per the instructions in the Installation chapter of this document.
- Install Reciprocal Net site software 0.9.1 and all supporting software packages per the instructions in the Installation chapter of this document. However, do not initialize Reciprocal Net or synchronize yet.
- Restore Reciprocal Net data from your backup, merge configuration files, update database schema, and synchronize per the *0.9.0 User Guide*.

Feel free to contact Reciprocal Net technical support for assistance with such an upgrade.

Backing up Reciprocal Net data

From the GNOME desktop, go to a command prompt and type the following:

```
recipnet-backup
```

. A backup program will halt some key daemons, back up the data for your Reciprocal Net site, and create an archive file in the `/root/` directory. This may take a few minutes depending upon the quantity of Reciprocal Net data on your system. Status updates are written to the console. Look for the “complete” message.

Store the archive file in a safe place. It will be needed in case this upgrade procedure goes awry.

The *mysqld* daemon was halted by the backup program run a moment ago. It must be started again. From the GNOME desktop, navigate to the *System/Administration* menu in the upper-left corner of the screen and launch the *Services* program. In the *Service Configuration* window that appears, look in the left-lower pane and find the *mysqld* item. Highlight *mysqld* and click the *Start* button on the program’s toolbar. Click the *OK* button on the success message that appears a few seconds later. Set the *Service Configuration* program aside but leave it open.

Reconfiguring yum

During the lifetime of release 0.9.0, several system administrators reported difficulty using *yum* to keep their operating system software up-to-date. *yum* frequently encountered conflicts between operating system packages and packages distributed by the JPackage project. Thus, the Reciprocal Net project now recommends a new configuration for *yum* to avoid similar difficulties in the future.

The first step is to update the repository definitions used by *yum*. Go to a command prompt and type the following:

```
wget -m -nd -P /etc/yum.repos.d/  
www.reciprocalnet.org/download /recipnet.repo
```

. Then, type the following command (all on one line):

```
wget -m -nd -P /etc/yum.repos.d/  
www.reciprocalnet.org/download /recipnet-jpackage-rhel5.repo
```

.

The second step is to verify your system has the most recent version of *yum*. From the GNOME desktop, navigate to the *Applications/System Tools* menu and launch the *Software Updater* program. The system will trundle for a moment and then display a list of all available updates. Select *yum* and *yum-rhn-plugin* if they are present; unselect all other packages that may appear. It is important that you do not attempt to update the other programs on your computer just yet. Click the *Apply updates* button. A progress

bar appears and packages are downloaded. If additional packages are required due to dependencies, click the **OK** button. Then Close the *Software Updater* program.

The next step is to install the *yum-priorities* plugin. From the GNOME desktop, navigate to the *Applications* menu and launch the *Add/Remove Software* program. In the window that opens, click the *List* tab. The program will trundle for a moment and then display a list of all software packages available for download. Scroll through the list; find and select *yum-priorities*. Click the *Apply* button. In the *Package selections* window that appears, click the *Continue* button. A *Downloading packages* window will appear with a progress bar. When it finishes, close the program.

Next, on Red Hat Enterprise Linux, from the GNOME desktop, navigate to the *Applications/Accessories* menu and launch the *Text Editor* program. In the window that opens, click the *Open* button. Navigate to the */etc/yum/pluginconf.d/* directory and open the file named *rhplugin.conf*. Edit the contents of the *rhplugin.conf* file to enable repository protection. Find the section of the file near the bottom, beneath the heading *[rhel-i386-server-5]*. In that section, insert the following text on a new line:

```
priority = 1
```

. Then click the *Save* button.

TIP→ If the *rhplugin.conf* file is present on your system but you do not see a *[rhel-i386-server-5]* heading inside it, this indicates your system does not have the proper version of the *yum-rhn-plugin* package. Version 0.5.30 or higher is required. Return to the *Software Updater* program and repeat the directions above.

TIP→ On some GNU/Linux distributions, there may be no file named *pluginconf.d*. Instead, check the directory named */etc/yum.repos.d/* for repositories associated with your operating system distribution. Open those repository configuration files and add *priority* directives (as above) to each repository declaration.

Next, in the *Text Editor* program window, click the *Open* button. Navigate to the */etc/yum/plugins.d/* directory and open the file named *priorities.conf*.

Edit the contents of the *priorities.conf* file to enable enhanced obsoletes checking. Scroll to the bottom of the file, in the section labeled *[main]*, and add the following text on a new line:

```
check_obsoletes = 1
```

. Then click the *Save* button. Close the *Text Editor* program.



Upgrading and reinstalling Reciprocal Net

In order to resolve the friction between operating system packages and JPackage packages, some packages will need to be removed from your system and reinstalled. These include Reciprocal Net site software and several dozen supporting packages. Going forward, due to the new *yum* conventions just configured, package conflicts should be far less common.

The first step is to remove several particular packages that may have been distributed by JPackage project. From the GNOME desktop, navigate to the *Applications* menu and launch the *Add/Remove Software* program. In the window that opens, click the *List* tab and let the program trundle for a moment. In the tab that appears, click the *Installed packages* option. Scroll to find the package named *jpackage-utils* and clear the check box beside it. Then scroll to find the packages whose name begins with *tomcat-...* and clear the check boxes beside those. (There may be several *tomcat-...* packages.) Click the *Apply* button. A *Dependencies added* window will appear to list a number of additional packages that will be removed. Verify that *recipnet-site-server*, *recipnet-site-utils*, and *recipnet-site-webapp* are included among them. Then click the *Continue* button. An *Updating software* button will appear with a progress bar. When the completion message appears, click the *OK* button.

The next step is to re-install Reciprocal Net site software and its supporting packages, downloading the newest releases of everything. Back in the *List* tab of the *Package Manager* program, click the *Available packages* option. Scroll to find the packages named *recipnet-site-server*, *recipnet-site-webapp*, and *recipnet-site-utils*; place check marks to each of the three. Then click the *Apply* button. In the *Package selections* window that appears, click the *Continue* button. A *Downloading packages* window will appear with a progress bar. When it finishes, close the program.

Downloading GNU/Linux updates

Chances are good that the preceding steps cleared several package dependency hang-ups on your system. Some operating system updates that failed to install on your system previously may install now.

To update your computer's software packages, from the GNOME desktop, find the *Applications* menu in the top-left corner of the screen. Navigate to the *Applications/System Tools* menu and launch the *Software Updater* program. The computer will trundle for a moment and then display a list of updates available. Verify that all of the available packages are selected. Click the *Apply updates* button. A progress bar appears and packages are downloaded.

If a *Reboot recommended* window appears, click the *Reboot now* button. Wait while the computer reboots. Repeat these steps – rebooting and running the *Software Updater* – until there are no more updates to be installed.

Merging configuration files

Some customizations you may have made to your Reciprocal Net installation might need to be recreated because the Reciprocal Net packages were uninstalled and reinstalled above. There are several configuration files to be examined.

Amongst these configuration files is *recipnetd.conf* that controls the *recipnetd* daemon. This file may or may not have contained customizations applied previously by the system's administrator.

The present release of the Reciprocal Net site software includes a new *recipnetd.conf* with a few new and updated configuration directives. These new directives are required for proper operation of site software 0.9.1. It is necessary to *merge* your new system's *recipnetd.conf* file with your old system's *recipnetd.conf* file. There is a utility for doing just that. At a command prompt, type:

```
recipnet-mergeconfig
```

. The configuration files are merged and several lines of descriptive output are produced. Leave this output open for a moment.

Now go to the GNOME desktop and double-click the *Computer* icon in the upper-left corner of the desktop. In the window that appears, double-click the *Filesystem* icon. Continue navigating to the *etc* folder and then the *recipnet* folder beneath it. Notice there are several files in the */etc/recipnet/* folder, including these three:

- *recipnetd.conf* , which is a default configuration file optimized for site software 0.9.1.
- *recipnetd.conf.rpmsave* , which is the configuration file that was saved from your previous installation.
- *recipnetd.conf.merge* , which is a merged version of the two files above.

Double-click the *recipnetd.conf.merge* file to open it in a text editor. In a moment, you will activate this file as your new system's configuration file. First, peruse the file and verify that all of your customizations (if any) were copied over from your old system correctly.

Review the output of the merge program in the terminal window. You should see a *Properties changed* section with one or more configuration directives listed beneath it. This means that for those directives listed, the merge program ignored the value that was found in *recipnetd.conf.rpmsave* (from your old installation), instead overriding it with a default value that is likely more appropriate for your new system running site software 0.9.1. Usually the merge program is correct, but you might wish to double-check these directives by looking at the configuration files with your text editor. If needed, refer to page 56 for a complete reference on the *recipnetd.conf* file.

Also output by the merge program in the terminal window should be a *Properties removed* section with zero or more configuration directives listed beneath it. The directives listed in this section were found in the *recipnetd.conf.rpmsave* (from your old



installation) but could not be copied automatically to *recipnetd.conf.merge* . They probably represent important customizations by your site's system administrator, such as locally-defined custom metadata files. If there are any configuration directives listed in this section, you will need to copy them manually. Open the *recipnetd.conf.rpmsave* file in a text editor window and find the named directives. Highlight these lines and copy them to the clipboard. (Look inside the *recipnetd.conf.rpmsave* file using a text editor; do not rely on the output of the merge program.) Then, open *recipnetd.conf.merge* in a text editor window, find the corresponding section of the file, and paste in the configuration directives.

When you've finished reviewing *recipnetd.conf.merge* and are ready to test it, save the file and exit the text editor program.

In the GNOME folder window, find the file named *recipnetd.conf* and right-click its icon. In the menu that pops up, select *Move to Trash*.

Then find the file named *recipnetd.conf.merge* . Right-click its icon and select *Rename* from the popup menu. Rename the file to *recipnetd.conf* . Right-click on the same file again and select *Properties* from the popup menu. In the window that appears, click the *Permissions* tab. Set the *Owner* to *recipnet*, the access beneath it to *Read and write*, the *Group* to *recipnet*, the access beneath it to *None*, and the access for *Others* to *None*. Then click the *Close* button. The merged configuration file now is active.

It is necessary also to remove (or at least rename) the *recipnetd.conf.rpmsave* file. Otherwise, future upgrades to the Reciprocal Net site software might be confused by the presence of this file and would attempt to merge configurations again. Right-click on the icon for *recipnetd.conf.rpmsave* and, in the menu that pops up, select *Move to Trash*.

Other configuration files to consider are those sitting in the */etc/recipnet/* directory. Any customizations not migrated above will need to be migrated manually. In particular, consider the *sitesponsor.gif* image file, which is the banner image displayed at the top of every web page served by your system.

Upgrading database schema

The MySQL database files used by Reciprocal Net site software have changed a bit in the new release. Update your system's database to the new schema by going to a command prompt and typing the following:

```
drecipnet update
```

. The program will output several lines of text and eventually a *Done* message. (If the *Done* message does not appear, this indicates a serious problem at your site that may result in data loss. Please do not proceed; instead, contact Reciprocal Net technical support.)

Synchronizing with the Reciprocal Net site network

The next step is to synchronize your server with the other servers in the Reciprocal Net Site Network. During synchronization, your server sends and receives messages over the Internet, processes any updates sent by the other servers, and tracks the samples publicized by the other servers.

This synchronization step is optional but recommended. If your server is up-to-date with respect to the Site Network then the process will finish in less than a minute. If your server has fallen out-of-date, however, now is a prime opportunity to remedy the situation.

Go to a command prompt and type:

```
drecipnet sync
```

. A bevy of status messages will be displayed, and any of them may seem to stall for several minutes. You'll know the process has concluded successfully when the word *Done* finally appears.

Wrapping up

By this point you should have upgraded all the necessary packages and site software 0.9.1 should be ready to run. If many packages were updated, and in particular if a Linux kernel was installed, a full reboot of the system may be prudent. Do this now.

If you prefer not to reboot completely, now is the time to start and re-start daemons. In the *Service Configuration* program from a few minutes ago, look in the left-lower pane and find the *recepnetd* item. Highlight it and click the *Start* button. Then find the *tomact5* item; highlight it and click the *Restart* button. Then find the *httpd* item; highlight it and click the *Restart* button. Then close the *Service Configuration* program.

And finally, once your daemons or your whole system has restarted, confirm that all your upgrades succeeded by visiting your server's web site (a URL something like <http://www.yourserver.edu/recipnet/>). The phrase *Reciprocal Net site software 0.9.1-50* should appear across the bottom of the page. Congratulations!



Using the Site Software

Terminology

It will be useful to become familiar with some basic terminology that will be used in this chapter.

A *sample* is the basic data entity in this system; this is what the Reciprocal Net site software tracks. Crystal samples usually are grown by one particular *sample provider* (the lab of a research chemist) and subsequently submitted to one particular *crystallography lab* for processing.

Structures may be contrasted with samples in that while one sample always illustrates one molecular structure, it is conceivable that two samples in Reciprocal Net would have the same underlying molecular structure. Generally, a *duplicate flag* is added to both samples when such a condition is discovered.

Now that we've already used the term we might as well define it: a *lab* is a participating crystallography laboratory that uses the Reciprocal Net site software for its daily operations (like you probably hope to do). Most management features of the Reciprocal Net Site Network take place at the lab level. For this reason, each sample has a data field where the originating lab is recorded at the time of sample creation.

A *provider* is generally the lab of a research chemist that grows crystal samples and submits them to your lab for analysis. There frequently are multiple people associated with a single provider, perhaps one faculty member and several graduate students. Each sample has a data field where the originating provider is recorded at the time of sample creation.

A *site* is a server (or cluster of servers) in the Reciprocal Net Site Network. More specifically, a site is a web server that runs the Reciprocal Net site software and is connected to the Internet. It is possible for a single site to host multiple labs. All sites in the network communicate with one another over the Internet to keep the distributed database of publicly visible samples in a consistent and accessible state. Sites also maintain their own collection of private samples that are not shared with other sites.

The *site database* is a SQL database that exists at each site and that contains metadata about samples in the Reciprocal Net Site Network. The database contains records for every sample that exists on the local site, plus those samples at other (remote) sites that have been made available to the public.

The *site repository* contains actual data files for samples in Reciprocal Net and is stored on the server's file system directly. These data files might include *.cif* files, *.sdt* files, *.ort* files, *.crt* files, *.pdb* files, and so forth. To recap: the site database contains *metadata*, the site repository contains *data*.

Initial administrative tasks

In a new Reciprocal Net site software installation, it's necessary to perform a few administrative tasks before samples can be submitted. Open a web browser and go to



<http://www.yourserver.edu/recipnet/> . (You may open the web browser on any Internet-connected computer; it need not be your Reciprocal Net server.) Log in with the *site administrator* credentials you specified in the previous chapter. A few new choices should appear on the web app's menu bar. Notice also that the name of your user account appears on the right side of the menu bar as a reminder to you.

Click on ***Administrative Tools*** to go to the main page for administrative tasks. From the screen that appears, click on ***Manage Labs***. (The ***Manage Labs*** link is the only link on that page that is functional in this release.) You should see a list of every lab in the Reciprocal Net Site Network. Click the link next to your lab that says ***Edit this lab***.

The top portion of this page contains some technical information about your lab record that you probably won't ever need to change. The bottom portion of the page contains a list of every user account that is associated with your lab. (In a new site software installation, the only account in the list will be the *site administrator* account that was created in the previous chapter.) Normally you will want to create individual user accounts for every employee who works in your crystallography lab. The way to do this is to click the ***Add a new user*** link.

In the form that appears, type the user's full name, username, and desired password. The latter two fields are limited to eight characters. You should be sure to set the user's ***Global access level*** appropriately. For example, crystallographers who work in your lab probably should be ***Scientific users***, while secretaries whom you enlist to help run your lab probably are not. You should think carefully before delegating ***Lab administrator*** and ***Site administrator*** privileges to others: while there is no limit to the number of each type of user that can exist, users with these privileges will have access to the ***Administrative Tools*** module that you're using right now. This means that they could create new user accounts or even deactivate your lab inadvertently. The only difference between a ***Site Administrator*** and a ***Lab Administrator*** is that the latter can administer only his own lab, while the former can administer any lab that's hosted on the site. Click ***Submit*** when you're done filling out the form. The newly-created user account should now appear on the list of users associated with your lab. Repeat this procedure until all your lab employees have their own user accounts.

The next step is to create *provider records* for every person or group that submits crystal samples to your lab. A single provider must be associated with every sample that you create in Reciprocal Net, and you must create at least one provider record before you can create new samples. If you aren't prepared to key all your providers into the system right now, just create a provider record called "Miscellaneous" or something similar.

To create a new provider, click ***Administrative Tools*** in the menu bar, then ***Manage Labs***, and then ***Manage this lab's providers***. (Provider records only make sense when considered in the context of the lab that created them; this is because providers do not have direct involvement with Reciprocal Net in the same way that labs do.) The list of providers associated with the lab should be displayed; it will be empty if this is a new installation. Click ***Add a new provider***. Name the provider something informative (like "Dr. Laue's group") and specify the head contact (Max Laue) in the next box. If your provider consists of only a single person, and not an entire research group, then it probably would make sense for you to leave the ***Provider Head Contact*** field blank. Enter any comments you like and then click the ***Submit*** button.

Now go back to the list of providers associated with your lab and click the *Edit this provider* link. You should see that data you entered before, and at the bottom of the form a list of user accounts associated with the provider. Creating user accounts for a provider is an optional step. You might do it if you wanted the sample provider to be able to access on your server the samples he submitted, giving him a quick and relatively painless link to the data files you created on his behalf. It's your choice whether you create a single user account that is shared by every person in the provider's research group, or whether you create individual user accounts for each member of the group.

When you're all done, click *Home* to conclude your tour of the *Administrative Tools*.

Creating a new sample

You should already have logged in to the web application and created a provider record, as was described in the preceding section.

The first step in creating a new Reciprocal Net sample is to create its metadata record using the web application. Click the *Submit Sample* button on the menu bar to go to the submission form. Notice that the lab with which your user account is associated is being recorded in the sample's metadata. In the next field you should choose the provider that delivered this crystal sample to you. You may create new provider records, if needed, via the *Administrative Tools* feature that was described above. A sample's associated provider cannot be changed once the sample record has been created. You may enter the sample's anticipated empirical formula, if the provider gave you some idea what it might be. You may also enter the name of the person within the provider's lab who actually delivered the sample to you.

Every Reciprocal Net sample must be assigned a unique *sample number* that identifies the sample within the context of your lab. It is not necessary for your sample number to be globally unique (across all of the Reciprocal Net Site Network), just for it to be unique within your lab. The sample number may be up to 32 characters long and may contain letters and symbols if you like. Many crystallography labs prefer to use a separate system for tracking samples submitted, using their own ID numbers. If your lab fits into this category you'll want to specify that tracking number here. Or, you could configure your server to auto-generate sequential sample numbers according to rules you specify. (See page 51.) It is very not possible to change a sample number once the sample record has been created, so choose wisely. Enter any comments you like and then click the *Submit* button.

Your browser should be delivered to the *Edit sample* page, where you can see details about this sample record and make changes to it. The next step in creating a new sample is to create its data directory in the repository; it's easiest if you do this via the web application. Scroll to near the bottom of the *Edit sample* page, in the section that begins with a message something like *No files for xxx in repository*. Every sample should have a corresponding directory in the repository where the sample's data files can be stored. In a default site software installation, the files are stored somewhere beneath */var/recipnet/data/*, usually under a directory named for the sample's originating lab and then under a directory named for the sample number. Every sample has its own data



directory, and this directory name must match the sample number. Some labs find it more convenient to organize their sample data directories into deeper hierarchies (perhaps with sample directories grouped into parent directories according to the year they were created, for example). This is an optional feature, but if you choose to use it you'll need to specify the "grouping directory" name in this box. If you do not wish to use this feature, and you're satisfied with all your sample data directories living in the same parent directory without any additional hierarchy, you may safely leave this box blank. Click the **Create** button to continue. The section of the **Edit sample** page that displayed a warning message before now contains a list of all the data files associated with this sample. No files will appear in this section, however, until you upload them to the server.

After you have submitted a new sample record and created its directory in the repository, the next thing you do to a real-world crystal sample is put it on your diffractometer and collect some data. Once data collection has completed, you'll want to return to the **Edit sample** page in the web app and click the **Collect preliminary data** link in the menu at the bottom of the page. The **Collect preliminary data** page gives you an opportunity to record information garnered from the collection operation – all fields are optional. You may specify the name of the crystallographer who collected the data (if more than one crystallographer was involved you may enter several names), the unit cell parameters (these can be changed later), the temperature at which data was gathered, a short (one-word) description of the crystal's color, and general comments about the collection operation. Click the **Save** button when you've finished filling out the form.

Once data has been collected from the real-world crystal sample and the operation recorded within the recipnet web app, the next step is usually to solve the structure and refine it. Unfortunately Reciprocal Net cannot help you with this part. But once the structure has been refined, be sure to visit the **Edit sample** page and click the **Refine structure** link. The **Refine structure** page allows you to record more information about the sample – again all fields are optional. The unit cell parameters specified on the **Collect preliminary data** page previously may be changed here; you may enter a summary of the structure if you wish; you may enter z and the space group, several statistical indicators of your data's accuracy (goodness of fit, RF, $R(F^2)$, RWF, $RW(F^2)$), and an empirical formula, structural formula, moiety formula, SMILES code, trade name, common name, and IUPAC (formal) name. Enter information in as many of these fields as you're able and then hit the **Save** button.

Your sample record is now in the **Complete** state and might stay this way for many months or years. The sample's provider, if he has a user account on your site, will be able to look his sample up and access the data you collected, as will fellow staffers who at your crystallography lab. Now is a good time to attach data files to your sample – like an industry-standard **cif** file or a **crt** file so that JaMM will work. All data files associated with samples that you create are stored on your server in the **repository**, somewhere beneath the directory **/var/recipnet/data/**. For instance, if you gave your sample the number '123' and the name of your lab was 'abc', and you chose not to use a "grouping directory", all the sample's data files should be placed in the directory **/var/recipnet/data/xyz/123/**. This directory should have already been created when you clicked the **Create** button earlier in this section. (Due to potential permissions and security issues, the site software web application is the preferred method for creating repository directories.)

You may upload data files to the server from your desktop computer via any of several mechanisms. Click the **Upload files** button near the lower-left corner of the page. You'll see the **Upload Files** page, where you can upload up to five files from your desktop computer to the server. Click a **Browse** button, choose a file from your local system, and click the **Upload now** button. The file is uploaded and a confirmation page appears. Subsequently you see the Manage Files page, which lists all sample data files attached to the sample and can manipulate them as need.

Another method for uploading files is via dragging-and-dropping to the web application. This is particularly effective if you have many files to transfer at once. On the **Manage Files** page, look in the lower-right corner and click on the link for **Upload files (Drag-and-Drop)**. The next page that appears contains a Java applet that helps you upload files. (If you see a security warning, please authorize the applet to run.) Arrange the windows on your screen so that you see both the web page and your local computer's file manager at the same time. Find a sample data file that you wish to upload within your desktop computer's file manager. Click on a file icon, drag the file into your web browser window, and drop the file onto the applet. A progress bar appears and the file is uploaded to the server. You can repeat the procedure for several files. You also can select several files on your desktop and drag all of them to the applet at once. When you've finished, click the **Save** button. A moment later you should see the **Manage Files** page again. Look in the lower-right corner and click the link **Back to "Edit Sample"**. Notice that links to the sample's attached data files have appeared on the bottom-left side of the page.

There are other ways to transfer sample data files to your server as well. Reciprocal Net site software automatically detects and recognizes sample data files that are deposited into existing repository directories on the server's filesystem. Thus, services like **ftp**, **ssh** (and related utilities), **nfs**, **samba**, etc. are useful for this purpose. Pretty much any of these mechanisms will require that the user transferring data files have a GNU/Linux user account on the server and write permissions on at least some portion of the repository.

The last step in dealing with a Reciprocal Net sample is releasing it to the public. This is the chief motivation behind the Reciprocal Net site software – making it as easy as possible for you to contribute your samples to the distributed public collection. In deference to intellectual property issues, most labs choose to publish samples on Reciprocal Net only after the same structure has been published in a trade journal (usually by the sample's provider), possibly several years after the Reciprocal Net sample reached the *complete* state. This ensures that the prior-publication rights of the sample provider are not infringed. Reciprocal Net also encourages its member labs to institute a policy that samples that have gone unpublished for five years or more (for example) are automatically released to the public by the crystallography lab staff.

To release a sample to the public, go to the **Edit sample** page and click the **Release to public** link. The **Release to public** page allows you to enter a few additional metadata items (all optional): a layman's explanation of the sample's real-world significance, a preferred name if there are several from which to choose, a copyright notice for your data and metadata, and comments about why the sample is being released to the public. The layman's explanation is particularly valuable if you intend to contribute your sample to the "Common molecules collection" maintained by Reciprocalnet.org but may be of limited use otherwise. The copyright notice is optional, but if you find your lab's



employees typing the same notice for many samples, you might want to go to *Administrative Tools* and *Edit lab* to change the default copyright notice that appears on samples released from your lab. Click the *Submit* button to release the sample to the public.

Unauthenticated visitors to your web site now will see the sample in the searches they run, and your sample will be indexed by the distributed search engine on Reciprocalnet.org very soon. Congratulations on releasing your first sample.

Importing metadata from CIF files

The *recipnet-site-utils* package includes a graphical utility for creating sample metadata records from existing *cif* files that you might have. This normally is most useful when you're first setting your site up and want to import "legacy" data.

TIP→ This release of the Reciprocal Net site software contains a web-based function that synchronizes a sample's metadata against the contents of a *cif* data file. Use of the web-based tool is recommended in most cases. By contrast, the utility described in the following section is most useful if you have dozens or hundreds of *cif*'s to be imported in-bulk to your server.

To run the utility, you'll need to be sitting at your server, be logged in as *root*, be running an X/Windows session (you might need to type *startx* at the command line to launch one), and have the *cif* file(s) sitting someplace in your server's filesystem. From a command prompt inside an X/Windows terminal, type:

```
recipnet-cifimporter filename.cif
```

where *filename.cif* is the path to the first of your *cif* files that you wish to import.

A window will appear with a list of data blocks found in the *cif* file in the left pane, a list of parse errors encountered in the *cif* file in the bottom pane, and import details about one particular data block in the right pane. (Normally a single *cif* data block will translate directly into one Reciprocal Net sample.) Click on the name of a data block in the left pane. Information extracted from this *cif* data block will appear in the right pane, as it would be translated to Reciprocal Net sample metadata. You can make any necessary additions to changes to the metadata record if you wish. In particular, you'll always need to check the *Lab*, *Provider*, and *Local Sample ID* (i.e. sample number) fields – these data items are an integral part of Reciprocal Net's tracking mechanisms and cannot be changed later. When you're satisfied with the information in the right pane, hit the *Store* button to create the metadata record in the database. Reciprocal Net samples created in this way automatically receive the *complete* state. You may continue importing *cif* data blocks (if there were more than one in your file) or simply exit the program.

After you've created a metadata record for your sample, you'll probably want to create a data directory in the repository and copy data files for the sample into it. You should use

the web application to create a directory in the repository for your sample as was described in the previous section. Then, simply populate the new directory with the sample's data files (including, possibly, the *cif* file you just imported) – the site software will detect and activate your data files immediately.

An alternate approach when importing a batch of *cif* files is to create repository directories first, populate them with *cif* files (and possibly other data files), and then run the *recipnet-cifimporter* utility to create the metadata records afterwards. Such an operation can be perilous and you should attempt it only if all of the following conditions are met:

1. Every one of your *cif* files contains exactly one data block.
2. You already know what sample numbers you plan to assign to each sample (and *cif* data block).
3. You do not plan to use “grouping directories” in your repository.
4. You feel comfortable manipulating GNU/Linux file and directory permissions.

In this alternate approach, the first step is to create individual repository directories for each *cif* file that you have. Don't forget: each directory name must match exactly the sample number you plan to assign later. If the samples all belong to the lab called 'abc', then the data directory you create for sample '123' should go in */var/recipnet/data/abc/123/*, without any extra grouping directories. Place one *cif* file in each new repository directory that you created, along with any other data files that you intend to attach to the samples. Now set appropriate permissions on all the new repository directories and files within them using GNU/Linux's *chmod*, *chown*, and *chcon* commands. Once this is done, go ahead and invoke *recipnet-cifimporter*. If you like, it is possible to specify several *cif* files on the command line by separating them with spaces. Create metadata records in this way (as was described earlier) for every *cif* file that you're importing. Finally, you need to link the repository directories you created manually to the metadata records you just created. Use the recipnet web application to go to the *Edit sample* page for each newly-imported sample. Near the bottom of the page, where attached data files normally appear, should be a message that says **No holding record**. Hit the **Create** button (without entering anything into the grouping directory box) to link your manually-created repository directory. When the page reloads, the data files you placed in the repository directory should appear. The import process is complete once you've done this for each new sample.

Attached data files

A *complete* sample that has been given number 123 should have two attached data files (at minimum): *123.cif* and *123.crt*. You are free to attach as many data files to each sample as you wish – some people include *gif* images, *ort* files for ORTEP renderings, *pdb* files, and so forth. The *crt* file in particular has special significance: it contains coordinate data about the unit cell observed in this sample and is required for JaMM (a Java applet for molecular visualization) to function. You may include multiple *crt* files depicting alternate views of the sample for users to switch between if you wish; in such a case the *crt* file named *123.crt* would describe the default view.

About chemical formulae

Several kinds of chemical formulae may be attached to a sample record:

- **Empirical formula:** this is the observed empirical formula of the sample. It is suggested that every sample record contain one of these. Coefficients and subscripts may be expressed as whole numbers or as decimals.
- **Empirical formula – less solvent:** similar to the standard **Empirical Formula** above, this kind of formula is different only in that any solvent of crystallization has been removed from the formula. Thus, only the “pure” formula of the substance is left. Such a formula is useful mainly for samples that are intended for release to a “general science” audience, such as those in Reciprocalnet.org’s Common Molecules collection.
- **Empirical formula – single ion:** similar to the standard **Empirical Formula** above, but this kind of formula represents only one (presumably complex) ion in a salt. Like the **Empirical formula – less solvent**, such a formula is useful mainly for samples that are intended for release to a “general science” audience.
- **Empirical formula – derived:** like the **Empirical formula – less solvent** above, this kind of formula is not a true empirical formula and is useful mainly for samples that are intended for release to a “general science” audience.
- **Structural formula:** a chemical formula written in a way that reflects some of the molecular structure details. Structural formulae should represent exactly the same cumulative element counts as the empirical formula for the sample (whether or not an empirical formula item is included among the sample data).
- **Moiety formula:** a chemical formula that describes the way the empirical formula is divided among discrete, covalently-bonded chemical moieties, but without specific structural detail. Moiety formulae should represent exactly the same element counts as the empirical formula for the sample (whether or not an empirical formula item is included among the sample data).

There is no limit enforced as to the number of chemical formulae of each type that may be attached to a sample record, but in most contexts it makes sense for there to be only a single formula of each type. If there are multiple formulae of a given type then Reciprocal Net reserves the right to discard all but one during certain operations not yet defined.

Reciprocal Net recommends that a specific chemical formula syntax be observed by all partner labs in order to facilitate reliable searching across various labs’ collections and improve the display of chemical formulae (formulae stored in standard format are displayed on web pages with appropriate subscript and superscript tags). Reciprocal Net

recommendations are based on CIF conventions (defined by the Crystallographic Information File format specification), including these rules:

- No chemical symbols other than recognized element symbols may be used.
- Each element symbol should be followed by a positive, numeric (not necessarily integer) count; a count of 1 may be omitted. (As specified below, a count of 1 *should* be omitted).
- A space or parenthesis must separate each cluster of element and count from any next and previous ones (but parentheses are valid only where specified below).

Going beyond the CIF conventions, Reciprocal Net recommends that these additional rules be observed:

- It is assumed that the numeric count for any element symbol or group is 1, unless the count is specified. A numeric count of 1 always should be omitted.
- Except in moiety formulae, any multiplier for a parenthesized group must follow the closing parenthesis (*i.e.* no pre-multipliers except in moiety formulae).
- Except in structural formulae, elements are listed according to the Hill system of Chemical Abstracts: if carbon is present then C should be first, followed by H, followed by all remaining elements in alphabetical order by chemical symbol; if carbon is not present then all elements should be in alphabetical order by chemical symbol. In moiety formulae this rule applies to each moiety individually.
- For those formulae that allow grouping (currently moiety formulae and structural formulae), the only characters that may be used for grouping purposes are parentheses.
- In addition, for moiety formulae:
 - Moieties are to be separated from each other by commas.
 - Parentheses may not appear inside moieties, but may surround them.
 - Applicable non-zero charges should be specified at the end of each moiety in the form of a + or – separated from the preceding element symbol and count by a space, and immediately preceded by the appropriate count if that is greater than 1.

About space group symbols

Each Reciprocal Net sample record may include the space group of the structure that was analyzed. Reciprocal Net recommends that a specific space group syntax be observed by all partner labs in order to facilitate reliable searching across various labs' collections and improve the display of space groups. Reciprocal Net recommendations are based on CIF conventions (defined by the Crystallographic Information File format specification).

It is expected that space group symbols entered by users be of the Hermann-Mauguin variety. Conformance with CIF requires that the full symbol (*c.f.* *International Tables for Crystallography, Vol. A* (1987)) be provided, and that the choice of origin be indicated if



it is not implicit. The lattice type symbol and symbols referring to different axes should all be separated by spaces, screw axes should be indicated by two consecutive digits, and alphabetic symbols should be given in their conventional case (*e.g.* “I 41/a c d”).

Reciprocal Net site software assists users in converting nonstandard space group symbols to standard format in some instances.

About units of measure

Scientific units of measure utilized by Reciprocal Net sample records conform to the standard CIF units (as defined by the Crystallographic Information File format specification), except for temperatures, which CIF specifies be in degrees Kelvin but Reciprocal Net stores in degrees Celsius.

Specifically, unit cell axis lengths are stored in *Angstroms*, unit cell inter-axis angles are stored in *degrees*, unit cell volume is stored in *cubic Angstroms*, density is stored in *grams per cubic centimeter*, chemical formula weight is stored in *grams per mole*, and temperature is stored in *degrees Celsius*.

Administering the Site Software



Web site customization

The Reciprocal Net site software's web application includes an image at the top of each web page that identifies the site's sponsor. Normally this is the logo of the crystallography lab that has joined the Reciprocal Net Site Network. You should replace this image file with one that includes the logo of your institution.

The image file is located on your server at `/var/www/html/recipnet/images/sitesponsor.gif`. Use the picture editing program of your choice to modify the file, but please maintain the existing image's size of 256 by 72 pixels. You can see examples of other site's logos by visiting <http://www.reciprocalnet.org/> and clicking on *Partner sites*.

There is no need to restart any daemons once you have replaced the *sitesponsor.gif* file. Apache Httpd makes the new file available to web browsers immediately.

The Reciprocal Net site software will not interfere with normal operation of the web site hosted on your server (by Apache Httpd). The site software web application will be accessible under the `/recipnet/` URL, but content placed anywhere else in your web tree should not be affected. Similarly, it is possible to modify Apache Httpd's configuration file to support other applications on your web site without interfering with the site software's operation.

Daemon startup order

The Reciprocal Net site software web application requires that no fewer than four system daemons be running in order for it to function properly. Relative startup order of these five daemons is important. The correct order, from first to last, is:

- *mysqld*, the MySQL database server daemon
- *recipnetd*, the Reciprocal Net site software daemon
- *tomcat5*, the Apache Tomcat servlet container daemon
- *httpd*, the Apache Httpd web server daemon

If you followed the instructions in this chapter to the letter then your computer is already configured to auto-start these services in the proper order. Recommended startup priorities for these four daemons are 64, 79, 80, and 85, respectively. Of course, the correct shutdown order for these daemons is the reverse of their startup order.

Ports and firewalls

It is assumed that each site in the Reciprocal Net site network will have an always-on connection to the Internet. The only port that needs to be open from the Internet to your server is TCP port 80 (HTTP), on which Apache Httpd answers incoming HTTP requests. Opening port TCP port 80 allows users across the globe to visit your web site

and access those samples that you have chosen to make public. Reciprocal Net site software does not require that any other port be opened.

Some system administrators find it convenient to enable SSH remote shell access to their servers. For this, TCP port 23 should be opened.

Good security practice dictates that all ports other than those listed above should be blocked. In particular, the following specific ports are in use by your server and are vulnerable to attack from the Internet unless blocked:

- TCP port 1099. This is used by *recipnetd* to receive queries from the web application running within *tomcat5*.
- TCP port 3306. This is used by *mysqld* to receive SQL database queries from *recipnetd*.
- TCP port 8005. This is used by *tomcat5* to receive shutdown signals from its daemon script.
- TCP port 8009. This is used by *tomcat5* to receive requests for JSP files that have been forwarded from *httpd*.

Red Hat Enterprise Linux includes a host-based firewall program called *iptables*. If you followed instructions in the Installation chapter to the letter, then *iptables* is already installed and configured properly. Further configuration can be effected from the GNOME desktop: navigate to the *System/Administration* menu and launch the *Security Level and Firewall* program.

For further information about firewalls and ports in use at your organization, please consult with your network administrator.

Backups

The wise system administrator keeps backups of all critical data on his server. When backing up all the data and metadata used by the Reciprocal Net site software, it is sufficient to preserve the files beneath */var/recipnet/* and */etc/recipnet/*. It may be prudent to stop the *recipnetd* and *mysqld* daemons for the duration of the backup so that no “dirty” files get backed up.

There are four directories under */var/recipnet/*:

- */var/recipnet/data/* - contains data files that have been attached to Reciprocal Net samples. This is the *repository*.
- */var/recipnet/db/* - used by MySQL to store its database files on behalf of *recipnetd*. These database files contain metadata about Reciprocal Net samples.
- */var/recipnet/msgs-held/* - used by *recipnetd* to store every inter-site message that it has received from other sites but that cannot be processed yet (for one reason or another). Files stored here are not critical because they



usually can be re-downloaded from other sites automatically by *recipnetd* if deleted.

- */var/recipnet/msgs-recv/* – used by *recipnetd* to store select inter-site messages that it has received. Not all inter-site messages received get written to files, but those that do are stored here.
- */var/recipnet/msgs-sent/* - used by *recipnetd* to store every inter-site message that it has sent (since the site was installed). Deleting, losing, or tampering with any files in this directory might result in serious data consistency problems across all sites in the Reciprocal Net Site Network.

To restore from a previous backup onto a server that already has the Reciprocal Net site software properly configured, it is sufficient to stop the *recipnetd* and *mysqld* daemons, replace all the files under */var/recipnet/* and */etc/recipnet/* with those from the backup, and then restart *mysqld* and *recipnetd*. If the server is not already properly configured, it will be necessary for you to install the most recent site software (a free download from <http://www.reciprocalnet.org>), place the *recipnet.sitegrant* file obtained from the Reciprocal Net Coordinator in */etc/recipnet/*, and modify */etc/recipnet/recipnetd.conf* to match your server's configuration.

Note that it is potentially disastrous to attempt to restore a *recipnet.sitegrant* file without the matching files in */var/recipnet*, or to attempt to restore the files in */var/recipnet* without the matching *recipnet.sitegrant* file. There are a number of interrelationships between the two entities; the risk is that other servers in the Reciprocal Net Site Network may become confused and cause your site to crash and lose data. If you should find yourself holding an incomplete backup of your server, please contact Reciprocal Net technical support for personal assistance.

Creating custom (locally-defined) metadata fields

Site administrators can define additional metadata fields that are attached to samples. Some labs find these fields convenient for storing information such as billing account numbers, requested completion dates, extra tracking information. Custom metadata fields are defined on a per-lab basis their visibility is configurable. They are not searchable and they are not replicated to other sites in the Reciprocal Net Site Network.

Each custom metadata field can store textual data up to 128 characters in length. Metadata contained within custom fields is visible in the recipnet web application on the *Sample Details* and *Edit sample* pages, and can be edited on most of the workflow action pages. Each site may define at most 1000 custom fields, with each field being used by a single lab.

Custom metadata fields are defined in *recipnetd.conf* by using the `SitLocalFieldxxx` directive, one metadata field per directive and one directive per line. The line in *recipnetd.conf* should look something like

```
SitLocalField000=12345,"Field name",65535
```

where **000** is any three-digit number uniquely identifying the field, **12345** is the numeric id of the lab that will use this custom field, and **Field name** is the human-readable name for the metadata field. The number **65535** is a bit mask that controls which web pages display the field (see below). There should be no spaces between the three directive parameters, and the field name should be enclosed in quotation marks. You may choose any three-digit decimal number for the directive, as long as each `SitLocalFieldxxx` directive in `recipnetd.conf` has a distinct number. The numeric lab id required by the directive can be found by using the recipnet web app, logging on as a site administrator, going to *Administrative Tools*, and going to the *Edit Lab* page for the desired lab.

In addition to identifying specific fields, the three-digit field numbers establish the relative order in which they are displayed by the recipnet web application: lower-numbered custom metadata fields are displayed before (higher on the page than) higher-numbered fields. Order of the directives in `recipnetd.conf` is not significant.

The visibility number is a bit mask that specifies which pages in the site software web application should expose the custom field. The minimum value is **0**, which means that the field is not exposed anywhere, and the maximum value is **65535**, which means that the field is exposed in all possible places. Other values require doing some binary math. The bits of the visibility number are defined as follows:

Bit number	Custom field exposed where?
(least significant) 0	Reserved
1	Submit page
2	Collect Preliminary Data page
3	Refine Structure page
4	Release to Public page
5	Modify Custom Fields page
6	Suspend page
7	Resume page
8	Declare Incomplete page
9	Failed to Collect page
10	Declare Non-SCS page
11	Retract page
12	Withdraw page
13	Sample Details page, visible to any authenticated lab user
14	Sample Details page, visible to any authenticated provider user
(most significant) 15	Sample Details page, visible to any unauthenticated user

After making changes to `recipnetd.conf`, you should restart `recipnetd` so that the changes will take effect. `httpd` and `tomcat5` should be restarted at the same time, as usual.

If you decide ever to stop using a custom metadata field, you may delete the appropriate `SitLocalFieldxxx` directive from `recipnetd.conf`. Keep in mind that this will not delete the actual metadata from the database, however, and therefore you should avoid using the same three-digit number on other custom fields in the future. A superior alternative might be to change the visibility number to **0**, which will cause the web application to not display the field on any page.

Configuring sample number auto-generation

Every sample record your lab creates within Reciprocal Net site software must have its own *sample number*, a 32-character alphanumeric label. This label is entered at the time each sample record is created. Some labs, as a matter of procedure, set this label in accordance with an existing, paper-based filing system. Other labs may prefer to have the sample number auto-generated in accordance with rules they set up.

In a default configuration, sample number auto-generation is disabled. System administrators can enable it by editing *recipnetd.conf* to include the configuration directives `SamLocalLabIdPrefixXXXXX` and `SamLocalLabIdAutoDigitsXXXXX`. For example:

```
SamLocalLabIdPrefix12345="prefix"  
SamLocalLabIdAutoDigits12345=3
```

where **12345** is the numeric id of the lab that will use this auto-generator, *prefix* is the sequence of characters that will be prepended to each sample number, and **3** is the number of digits to be included in each sample number. These example directives would lead to sample numbers *prefix001*, *prefix002*, *prefix003*, and so forth.

The numeric lab id required by the directive can be found by using the recipnet web app, logging on as a site administrator, going to *Administrative Tools*, and going to the *Edit Lab* page for the desired lab.

How your site interacts with the Site Network

When you install the site software RPM's on your web server, you join the Reciprocal Net Site Network, an Internet community of crystallography labs that donate their time and their servers to the project. The primary function of the Reciprocal Net site software you installed on your server is to act as a web application, allowing authenticated users to submit and alter sample data and metadata, and unauthenticated users to access data and metadata for a select set of public samples. This all takes place in the standard web application paradigm, where human users direct their web browsers to your web site.

The Reciprocal Net Site Network performs a considerable amount of behind-the-scenes management in order to keep its distributed database consistent. The several sites in the network exchange messages and files with one another and with the Reciprocal Net Coordinator on a fairly frequent basis. It is *recipnetd*, the Reciprocal Net daemon, which performs this function at your site. The remainder of this section will provide an overview of this interaction.

Your site will disclose sample data and metadata in the following circumstances:

- *when requested by unauthenticated users* via the web application, provided the sample has been **Released to Public**.

- *when requested by authenticated human users* via the web application. The authenticated user is authorized to access a sample if any of the following conditions are met: 1) the authenticated user's account belongs to the same lab from which the sample originated, 2) the authenticated user's account belongs to the same provider from which the sample originated, 3) the authenticated user's account has been authorized explicitly by an entry in the sample's access control list, or 4) the sample has been **Released to Public**.
- *when requested by a harvesting agent* elsewhere on the Internet via OAI (Open Archives Interface), provided the sample has been **Released to Public**. OAI is a protocol layered on top of HTTP that is commonly used in the digital library community for automated, distributed metadata harvesting. It generally does not require client authentication.
- *when requested by another site* in the Site Network via bulk HTTP file transfer, provided the sample has been **Released to Public**.
- Sample metadata is broadcast automatically to other sites (as described later) whenever a sample's metadata record is modified, provided the sample has been **Released to Public**.

It is anticipated that outside organizations like the National Science Digital Library (NSDL) and the Cambridge Structural Database (CSD) will catalog public samples in the Site Network in an accurate and reliable fashion using the interfaces described above. It is also conceivable that additional interfaces may be created for such purposes in the future.

Your site will receive, accept, and process signed messages from the Reciprocal Net Coordinator for the following purposes:

- *new site initialization* during the installation process, by way of the **recipnet.sitegrant** file installed on your computer.
- *maintaining a globally consistent list of sites*. A message is received whenever another site in the Site Network is added to the network, removed from the network, or information in its site record is modified.
- *maintaining a globally consistent list of labs*. A message is received whenever a lab in the Site Network is added, or when information in the lab record is modified by the Coordinator (lab modifications may be made also by the site on which the lab is currently hosted).
- *maintaining a globally consistent list of sample id's*. A message is received whenever the Coordinator transfers sample id's to other sites, or claims sample id's for itself (perhaps with the intent of transferring them to a new site).
- *collecting global statistics*. A message is received from the Coordinator whenever statistics are being requested from your site. This may occur periodically. In response, your site will send a message to the designated



statistics collection point in which certain operating statistics will be disclosed. This is discussed in more detail later.

Your site will broadcast information to all other sites in the Site Network for the following purposes:

- *maintaining a globally consistent list of labs.* An updated lab record is broadcast to the other sites whenever any information about a lab hosted at your site is changed, such as when an administrator uses the web app's **Edit Lab** feature.
- *maintaining a globally consistent list of providers.* A provider record is broadcast to the other sites whenever any provider associated with a lab hosted at your site is changed (such as when an administrator uses the web app's **Edit Provider** feature), or whenever a new provider is created (such as when an administrator uses the web app's **Add New Provider** feature).
- *maintain a globally consistent set of sample metadata for public samples.* A sample metadata record is broadcast to the other sites whenever a sample belonging to a lab hosted at your site becomes publicly visible (via the web app's **Release to Public** feature), loses its publicly-visible status, or a publicly-visible sample has its metadata updated. In the future, an updated sample metadata record may be broadcast to the other sites whenever data files in the repository for a publicly-visible sample are added, deleted, or modified.
- *maintaining a globally consistent catalog of sample data for public samples.* A sample data holding announcement is broadcast to the other sites whenever data for a public sample begins to be hosted at your site (such as when a new sample originated by a lab at your site is **Released to Public** via the web app, or public sample data arrives at your server by some other means). Another message is broadcast if the sample data later becomes unavailable.
- *maintaining a globally consistent directory of sample id's.* Sample id negotiation messages are broadcast to the other sites from time to time as your site generates new sample id's and coordinates with other sites as they generate their own new sample id's. A distributed algorithm is required for this process in order to ensure that the sample id's used by your site are globally unique across all of the Site Network. It is theoretically possible that a malicious person elsewhere on the Site Network could track these broadcasts to infer the number of samples (both public and private) that have been created at your site, to the nearest thousand. It would not be possible for the malicious person to determine the way in which sample id's were distributed among the individual labs at your site.
- in the future, *real-time network-wide load balancing.* General statistics regarding server/network load will be broadcast to the other sites periodically in order to facilitate load balancing across the distributed database. These statistics would be limited to data items like CPU load factor, number of simultaneous human web users (with no distinction made between those

authenticated and those not), number of samples and bytes served recently (with no distinction made between samples that are public and those that are not), observed throughput to other sites, and so forth. (As of the present release, the site software does not yet have this capability.)

Your site will send private messages to other sites in the Site Network for the following purposes:

- *negotiating sample id's* as was described previously.
- *reporting aggregate statistics*. Your site will disclose operating statistics (as described later) to the designated collection point in response to a signed request from the Coordinator.

Your site will disclose general statistics regarding its routine operation to the designated collection point in response to signed requests from the Reciprocal Net Coordinator. Such requests may be received on a regular basis. The statistics then are used by the Coordinator to publish whole-network utilization figures, track network growth, troubleshoot abnormal activity, and so forth. The statistics disclosed are:

- Total uptime and software version of *recipnetd*. These figures will not be published by the Coordinator in conjunction with any information that might identify your site.
- Total number of web app sessions, broken down by those in which users authenticated and those in which users did not. These figures will not be published by the Coordinator in conjunction with any information that might identify your site.
- Total number of “page views” on samples served via the web app, broken down by lab, and then broken down by visits to the *Show Sample* page versus visits to the *Edit Sample* page. These figures will not be published by the Coordinator in conjunction with any information that might identify your site or any lab at your site.
- Total combined size (in bytes) of all data files in the repository, along with the mean size of each data directory with figures broken down by lab. These figures will not be published by the Coordinator in conjunction with any information that might identify your site or any labs at your site.
- Total number of samples that have been released to the public, broken down by lab. This same information is already readily apparent to any unauthenticated user who browses Reciprocalnet.org; therefore, the Coordinator reserves the privilege of publishing this figure in conjunction with the name of your lab or site.
- Total number of messages exchanged with every other site in the Site Network, along with a list of each unique sequence number used. The actual contents of these messages are not disclosed. This information may facilitate debugging and troubleshooting. (The circumstances under which messages



can be exchanged were described previously.) These figures will not be published by the Coordinator in conjunction with any information that might identify your site.

- Total number of OAI queries that were answered. Also a count of the total number of samples returned by those queries. These figures will not be published by the Coordinator in conjunction with any information that might identify your site.
- Total number of HTTP bulk file transfer requests that were accepted. Also a count of the total number of files and the total number of bytes that were transferred during those operations. These figures will not be published by the Coordinator in conjunction with any information that might identify your site.
- In the future, the National Science Foundation (NSF) and/or the National Science Digital Library (NSDL) may request information about the Site Network's utilization that requires *recipnetd* to collect additional and possibly more specific statistics, and the Coordinator in turn to collect them. Such additional data collection activity would be well-defined and will not track web app visitors from visit to visit, track authenticated human users except in the aggregate, track the sort of detailed HTTP information normally written to a "web server log" except in the aggregate, or track information about non-public samples except in the aggregate and only when contrasted with public samples at the same site.

About *recipnet.sitegrant* files

The *recipnet.sitegrant* file is a binary configuration file that must be present on every site in Reciprocal Net. Every *recipnet.sitegrant* file is unique and custom-built for the site. It's not part of the standard software distribution – after you install the *recipnet* RPM on your server, you'll need to obtain a *recipnet.sitegrant* file from the Reciprocal Net Coordinator and place it in your `/etc/recipnet/` directory. Only the Reciprocal Net Coordinator can issue these *site grant* files.

The *recipnet.sitegrant* file contains important initialization data specific to your site. The most important elements are your site's unique identification number, your site's encryption keys, and the identity of the Coordinator. Reciprocal Net is a distributed system and therefore requires that id numbers of its data items be unique. Centralized creation of *recipnet.sitegrant* files by the Coordinator ensures that there are no duplicate site numbers, lab numbers, or sample numbers, and that every site knows how to communicate with every other site. It is important to understand that this file is *not* intended to be a licensing scheme of any sort – Reciprocal Net is 100% free and open.

A *recipnet.sitegrant* file contains a site's encryption key pair, although this might seem curious at first glance. Reciprocal Net sites are able to exchange messages with one another across the Internet, and for security reasons, cryptographic digital signatures are added to every message and verified by receiving sites. Encryption keys are necessary

for this to work. Reciprocal Net does not have mechanisms for encrypting data so that it cannot be read by others in transit (nor does it need them), only for signing messages so that they cannot be tampered with by crackers.

Your site stores its private encryption key in the file *recipnet.sitegrant*. Therefore, it is important that you secure this file and prevent unauthorized access. At minimum, you should set permissions on this file so that only the user *recipnet* can read it.

The *recipnet.sitegrant* file isn't obscured deliberately, but it does contain a lot of compressed binary data like encryption keys and so forth. There are no settings in this file that affect how your server behaves; all of Reciprocal Net's configuration parameters are stored in *recipnetd.conf*, which is human-readable.

Currently the Coordinator function is being performed by the Molecular Structure Center at Indiana University. To obtain a *site grant* file for your site, send an e-mail to Dr. Maren Pink (info@reciprocalnet.org) telling her that wish to turn your server into a Reciprocal Net site. She'll contact you offline (probably via telephone or videoconference) just to say hello and to arrange for the file exchange. The process normally takes a couple of business days.

We hope to create an automated mechanism at some point in the future for registering new sites and obtaining *recipnet.sitegrant* files.

Configuration directives in recipnetd.conf

recipnetd.conf is a text file that lives in */etc/recipnet* and that controls the operation of *recipnetd*, the site software daemon (also called the "core modules"). Directives are in key=value format, one per line. Blank lines and lines beginning with the # character are ignored. Order of directives is unimportant. Case in directive names (and usually in values) is important. Values should not be enclosed in quotation marks or any other delimiters, nor contain spaces.

Currently the following directives are supported:

DbDriverClassName	Java class name of the JDBC driver for your database engine that's contained in the JAR file specified by DbDriverFile . In a default installation (with MySQL as the database engine) this would be <i>com.mysql.jdbc.Driver</i> .
DbDriverFile	Complete file specification (path) to the JAR file that contains the code for the JDBC driver for the relational database engine that's in use. In a default installation (with MySQL as the backend database) this would be <i>/usr/share/java/mysql-connector-java.jar</i> .
DbUrl	The URI of the relational database engine, in a form that JDBC understands. In a default installation



(using MySQL as the database engine) the string should be

jdbc:mysql://localhost:3306/ recipnet .

In order, the components of the string are a standard method tag, the name of the database engine in use, the host name of the database engine, the port the engine is listening on, and the database (or “catalog”) name to be used. It is possible to cause *recipnetd* to connect to a database engine to a host other than the one on which *recipnetd* is running in this way.

DbUrlForBootstrap	Similar in form to DbUrl above, except this value is used only during so-called “bootstrap” operations like installing a new site or upgrading the site software from one version to another. In a default installation this value is <i>jdbc:mysql://localhost:3306/ recipnet .</i>
GenRecipnetdClassFile	Complete file specification (path) to the JAR file that contains the code for <i>recipnetd</i> . In a default installation this would be <i>/usr/share/java/recipnetd.jar.</i>
GenRmiPort	The TCP port number on which <i>recipnetd</i> is to listen for incoming connections from the site software web application hosted by <i>tomcat</i> . If both <i>tomcat</i> and <i>recipnetd</i> are present on a single server, it is recommended that this port be blocked in order to prevent other computers from interfering with this communication. The default value is <i>1099 .</i>
GenUtilityClassFile	Complete file specification (path) to the JAR file that contains the code for the various command-line utilities included in the recipnet package. In a default installation this would be <i>/usr/share/java/recipnet-utils.jar.</i>
handlers	Java class name of the “handler” that <i>recipnetd</i> ’s logging subsystem should use. In a default installation this should be <i>org.recipnet.site.core.SyslogHandler</i> – this handler sends major events to GNU/Linux’s <i>syslog</i> mechanism. It is possible to specify another class name here to cause log text to be written to <i>stderr</i> , for example.
IsmSuicideNote	Complete file specification (path) of a file that <i>recipnetd</i> will create to store debugging information in the event that a severe programming error is

encountered. The recipnet user must have write permission in the directory specified. In a default installation this would be */tmp/recipnet-ismSuicideNote.txt* .

- LogCommand** Complete file specification (path) to the command-line utility that *recipnetd* invokes to send log messages to syslog (or a similar mechanism). In a default installation this would be */usr/bin/logger*.
- LogFacilty** The name of the *syslog* ‘log facility’ to which log events should be written. The definitive reference for these is *syslog*’s man page, but possible values (at the time of this writing) are *auth*, *authpriv*, *cron*, *daemon*, *ftp*, *kern*, *lpr*, *mail*, *news*, *syslog*, *user*, *uucp*, and *local0* to *local7*. This should be *user* in a default installation.

This directive is ignored if *handlers* is not set to *org.recipnet.site.core.SyslogHandler*.
- LogFilter** This can be set to the Java class name of a ‘log filter’ to be used by *recipnetd*’s logging subsystem. The value of this directive should be blank in a default installation.

This directive is ignored if *handlers* is not set to *org.recipnet.site.core.SyslogHandler*.
- LogForceSynchronous** Set to *false* in a default installation, but may also be true. If *false*, log events are sent to GNU/Linux’s *syslog* facility asynchronously in order to improve performance. However, in the event of a *recipnetd* crash, it is conceivable that log messages might be lost. Set this to *true* to improve the chances that a log event would get written to *syslog* upon a crash of *recipnetd*.

This directive is ignored if *handlers* is not set to *org.recipnet.site.core.SyslogHandler*.
- LogFormatter** This can be set to the Java class name of a ‘log formatter’ to be used by *recipnetd*’s logging subsystem. Log formatters translate log events into plain text that is written to *syslog*. The value of this directive should be blank in a default installation.

This directive is ignored if *handlers* is not set to *org.recipnet.site.core.SyslogHandler*.



LogLevel	<p>The threshold that a log event (generated by <i>recipnetd</i>) must meet or exceed in order to be logged. Possible values are <i>SEVERE</i>, <i>WARNING</i>, <i>INFO</i>, <i>CONFIG</i>, <i>FINE</i>, <i>FINER</i>, <i>FINEST</i>, <i>ALL</i>, and <i>OFF</i>. The recommended value is <i>WARNING</i>. Note that it is possible to configure GNU/Linux's <i>syslog</i> mechanism to enforce its own threshold, independent of the one specified here.</p> <p>This directive is ignored if <i>handlers</i> is not set to <i>org.recipnet.site.core.SyslogHandler</i>.</p>
LogTag	<p>The textual 'tag' to use when sending log events to GNU/Linux's <i>syslog</i> mechanism. Tags can be useful when it comes time to analyze a combined log file generated by many different programs that all write to <i>syslog</i>. This is set to <i>recipnetd</i> in a default installation, but may be changed to any other word.</p> <p>This directive is ignored if <i>handlers</i> is not set to <i>org.recipnet.site.core.SyslogHandler</i>.</p>
RepBaseDirectory	<p>The base directory of the repository, the place where data files that have been attached to samples are stored. In a default installation this would be <i>/var/recipnet/data</i>.</p>
RepCvsCommand	<p>The path on the local filesystem to GNU/Linux's <i>cvs</i> program. In a default installation this is <i>/usr/bin/cvs</i>.</p>
RepCvsDirectory	<p>The directory within the local site's repository beneath which details about prior versions of sample data files should be archived. In a default installation this value is <i>/var/recipnet/data/cvs/</i>.</p>
RepDbPassword	<p>The password (in plaintext) associated with the database user account specified by <i>RepDbUsername</i>.</p>
RepDbUsername	<p>The name of the first of three database user accounts used by <i>recipnetd</i> to communicate with the backend database engine. (The other two usernames are specified by <i>SamDbUsername</i> and <i>SitDbUsername</i>.) The three user accounts specified should be distinct from one another in order to avoid concurrency conflicts, particularly with MySQL.</p>
RepHoldingIdsCache	<p>The first of several directives that control cache behavior inside <i>recipnetd</i> – tuning these values can have substantial performance implications both positive and negative. Every cache directive takes</p>

seven parameters on a line that looks something like:

RepHoldingIdsCache=32,128,8,8640000,56000,0.1,false

This particular directive controls a cache in Repository Manager that matches sample id's to the id's of potentially many holding records in the database.

The seven cache parameters, in order, are:

Cache size: the maximum number of items the cache is able to store at once. This must be larger than 0.

History log size: the maximum number of previous cache hits about which to keep statistics. Accurate statistics assist the cache in optimizing itself. Set this to 0 to disable hit logging (which can consume memory).

Cache chunk size hint: the approximate number of items that should be removed from the cache during each pruning operation. Larger numbers here will reduce the needed frequency of pruning operations, which can be expensive. This must be larger than 0.

Maximum age: the maximum number of milliseconds an item may live in the cache before it is considered to be expired and will be discarded. Set this to 0 to disable item aging.

History log reduction interval: the number of milliseconds between cache log reduction operations, where the oldest record in the cache hit log is removed. This feature can improve responsiveness of the cache's self-optimizing mechanisms during periods of low activity. The longest possible time an entry can live in the cache hit log is *History log reduction interval * History log size*. Set this to 0 to disable history log reduction.

Random factor weight: a number between 0 and 1 that indicates the extent to which a randomly-generated factor is used to influence decisions about which cache items are more valuable to retain than others during pruning operations. A random factor can be useful in improving performance when the working set is larger than the cache's size. If this value is 0.1, for example, then the relative scores of two cache items can be adjusted up to +/- 10%,



according to the randomly-generated number. Set this to 0 to negate the effect of the random number.

Exception: if history logging is disabled (by setting *History log size* to 0) then the Random factor weight has no upper bound. In this case it represents the maximum number of milliseconds by which the ages of two cache items can be adjusted during scoring.

Secondary key support: set to either *true* or *false*. In practice, secondary key support is necessary only for the `SiteUserCache` and should be disabled everywhere else for performance reasons.

`RepHoldingsCache`

Specifies parameters for a cache in Repository Manager that stores recently used “holding records” that have been read from the database. The suggested parameters are ***256,128,32,86400000,56000,0.1,false***. See `RepHoldingIdsCache` for a complete description of cache parameters.

`RepPriorVersionFileSizeLimit`

The maximum size, in bytes, of a sample data file for which unauthenticated users may request a prior version. Making available prior versions of very large sample data files consumes a great deal of server resources; thus, allowing unauthenticated users to do so might pose a security risk. The suggested value is ***8388608*** (8 megabytes). The limit is not enforced for authenticated users affiliated with the sample in question.

`RepPriorVersionTotalSizeLimit`

The maximum size, in bytes, of the “available prior versions of sample data files area” on the local filesystem. Sample data files are stored in this area temporarily to facilitate downloading, by user request, when they belong to a prior version of a sample. If this limit is exceeded, files are rotated out of the area on a least-recently-used basis. The suggested value is ***4294867296*** (4 gigabytes).

`RepRmiName`

A string that specifies the name with which Repository Manager registers itself to RMI. This value must match the corresponding entry in *web.xml* (used to configure the web application). In a default installation, this value is ***RecipnetRepositoryManager***.

`RepScannerTask`

The first of several parameters that affect how periodic tasks are executed within *recipnetd*; these can affect both performance and usability. Every task directive takes three parameters that looks something

like:

RepScannerTask=3600000,600000,3

The periodic task controlled by this particular directive scans the entire repository file system to look for newly-created sample data directories.

The three task parameters, in order, are:

Approximate interval: the approximate number of milliseconds that should pass between successive executions of the periodic task. It is not possible to schedule tasks for exactly periodic execution via this mechanism.

Maximum runtime: the maximum number of milliseconds this task might take to complete, under adverse circumstances. This information is used to detect stalled tasks.

Characteristic: an integer composed of several flags that help the scheduling engine decide which task to run when. Two tasks with the same characteristic bit set will not execute simultaneously.

RepScriptCreateDir	Complete file specification (path) to the command-line utility that <i>recipnetd</i> invokes to create new data directories in the repository. In a default installation this would be <i>/usr/bin/recipnet-createreposdir</i> .
RepScriptFixFile	Complete file specification (path) to the command-line utility that <i>recipnetd</i> invokes to fix permissions on newly-created data files in the repository. In a default installation this would be <i>/usr/bin/recipnet-fixreposfile</i> .
RepScriptLabDirSize	Complete file specification (path) to the command-line utility that <i>recipnetd</i> invokes to discover the total size of a filesystem directory. In a default installation this would be <i>/usr/bin/recipnet-getrepositorysize</i> .
RepUploadedFilePrefix	The prefix used to name temporary files created during uploads of sample data files. Each temporary file created lies in its associated sample's data directory. The suggested value is <i>.upload</i> . Because this value begins with a dot, on GNU/Linux this implies that all temporary upload files would be "hidden".



RepWorkerThreadName	A string that specifies the name that Repository Manager gives to its worker thread. In a default installation this value is <i>RepositoryManager</i> .
RepWorkerThreadShutdownTime	The maximum number of milliseconds Repository Manager's worker thread may take to shut down before it's considered to be stalled. This is <i>5000</i> in a default installation.
RepWorkerThreadStartupTime	The maximum number of milliseconds Repository Manager's worker thread may take to start up before it's considered to be stalled. This is <i>5000</i> in a default installation.
SamDbConnectionCount	The number of database connections that should be placed into the "sample- or repository-related" connection pool. Database connections from this pool are used to fetch samples and execute sample searches. This number governs the maximum number of "sample-related" operations your site will perform simultaneously and may have far-reaching performance implications. This is <i>8</i> in a default installation.
SamDbPassword	The password (in plaintext) associated with the database user account specified by SamDbUsername .
SamDbUsername	See RepDbUsername .
SamIdBlockProposalPeriod	The maximum number of milliseconds that the distributed algorithm used by your site to generate new sample id blocks takes to finish. The default value is <i>60480000</i> (7 days).
SamIdBlockTransferRequestPeriod	The maximum number of milliseconds that a sample id block emergency loan request generated by the local site may remain active. The default value is <i>60480000</i> (7 days).
SamIdMaintenanceTask	Specifies parameters for a task in Sample Manager that checks to see if your site owns a sufficient number of sample id blocks and takes action to acquire more if necessary. The suggested parameters are <i>21600000,60000,10</i> . See RepScannerTask for a complete description of periodic task parameters.
SamLocalLabIdAutoDigitsXXXXX	If present, enables sample number auto-generation for a particular lab. Replace XXXXX with the numeric lab id. Optional directive. Use in conjunction with SamLocalLabIdPrefixXXXXX . See page 51 for more

information.

SamLocalLabIdPrefixXXXXX	If present, enables sample number auto-generation for a particular lab. Replace XXXXX with the numeric lab id. Optional directive. Use in conjunction with SamLocalLabIdAutoDigitsXXXXX . See page 51 for more information.
SamLockLogStackTraces	Either <i>true</i> or <i>false</i> . Set this to <i>true</i> to enable detailed logging of <i>recipnetd</i> 's locking subsystem for debugging purposes. Application performance is much better if this value is <i>false</i> , the suggested value.
SamLockTimeout	The maximum number of milliseconds that a user request should be delayed when the resources required to service the request are locked by another user. Once this value is exceeded, the user receives a "deadlock" error. The suggested value is 60000 (60 seconds).
SamPreexecuteSearches	Either <i>true</i> or <i>false</i> . Set this to <i>true</i> to reduce the apparent latency when web users run sample searches.
SamRandomizerTask	Specifies parameters for a task in Sample Manager that seeds the random number generator every now and then. The suggested parameters are 300000,2000,0 . See RepScannerTask for a complete description of periodic task parameters.
SamRmiName	Similar to RepRmiName . In a default installation this value is RecipnetSampleManager .
SamSampleCache	Specifies parameters for a cache in Sample Manager that stores sample records that have been read from the database. The suggested parameters are 256,512,32,0,25000,0.1,false . See RepHoldingIdsCache for a complete description of cache parameters.
SamSearchParamsCache	Specifies parameters for a place in Sample Manager that stores criteria for searches that users have run. The maximum number of searches that can be active on a site at any one time is equal to the size of this cache. When an item is dropped from this cache (whether due to age or space limitations) the search is no longer available and the user would need to respecify his search criteria. The suggested parameters are 512,128,32,86400000,56000,0.1,false . See RepHoldingIdsCache for a complete description

of cache parameters.

SamSearchResultsCache	Specifies parameters for a cache in Sample Manager that stores search results for recently-run sample searches. For efficiency's sake, the size of this cache should be larger than the number of searches that are typically active on the local site at any time. The suggested parameters are 32,128,4,3600000,56000,0.1,false . See RepHoldingIdsCache for a complete description of cache parameters.
SamWorkerThreadName	Similar to RepWorkerThreadName . In a default installation this value is SampleManager .
SamWorkerThreadShutdownTime	Similar to RepWorkerThreadShutdownTime . This is 5000 in a default installation.
SamWorkerThreadStartupTime	Similar to RepWorkerThreadStartupTime . This is 5000 in a default installation.
SamUnusedIdBlockLoanThreshold	A whole number that defines the minimum number of free sample id blocks your site will try to hold at any given time. (Each sample id block contains 1024 globally unique sample id numbers that may be assigned to new sample records generated by your site.) When the number of free sample id blocks owned by your site drops below this number, your site will refuse to "loan" sample id blocks to other sites when they run low. The default value is 2 .
SamUnusedIdBlockMinimum	A whole number that defines the bare minimum number of free sample id blocks your site will hold at any given time. When the number of free sample id blocks owned by your site drops below this number, your site will begin an urgent process to "borrow" additional sample id blocks from other sites. The default value is 1 .
SamUnusedIdBlockTarget	A whole number that defines the maximum number of free sample id blocks your site will try to hold at any given time. When the number of free sample id blocks owned by your site drops below this number, your site will attempt to acquire more blocks by coordinating with other sites in a distributed algorithm. The default value is 4 .
SitHttpConnectTimeout	The maximum number of milliseconds Site Manager will wait for an acknowledgement after attempting to transmit an inter-site message to a remote site before

assuming that the remote site is down. The default value is **1000**.

SitHttpReadTimeout

The maximum number of milliseconds Site Manager will wait for a reply after requesting a “pull” of inter-site messages from the remote site. The default value is **60000**.

SitIsmProcessingTimeout

The maximum duration (in milliseconds) your site will spend attempting to process an inter-site message it received. The default value is 300000 (5 minutes). In practice, most inter-site messages are processed in less than 500 milliseconds.

SitLabCache

Specifies parameters for a cache in Site Manager that stores lab records that have been read from the database. For efficiency’s sake, the size of this cache should be equal to or greater than the total number of labs in the Reciprocal Net Site Network. The suggested parameters are **128,0,16,0,0,0,false**. See **RepHoldingIdsCache** for a complete description of cache parameters.

SitLocalFieldxxx

Replace **xxx** with any three-digit whole number from 000 to 999. Creates a custom field (or “locally-defined field”) for samples created by the specified lab. Lab users are able to attach extra textual metadata to samples in this way. Optional directive. See page 49 for more information.

SitMsgsAlwaysDump

Either **true** or **false**. Set this to **false** for performance reasons unless you’re running additional software at your site that piggybacks on **recipnetd**’s mechanism for exchanging inter-site messages.

SitMsgsHeldDir

Complete directory specification (path) to the place where **recipnetd** should store message files for the inter-site messages that it has received but that cannot be processed yet. The directory cannot be used for any other purpose and may not contain extraneous files. In a default installation this would be **/var/recipnet/msgs-held/**.

SitMsgsHoldTime

The maximum number of milliseconds during which Site Manager will hold and attempt to redeliver a received inter-site message that encounters transient processing errors. After this time has elapsed the message is dropped. The default value is **60480000** (7 days).



SitMsgsPush	Either <i>true</i> (the default) or <i>false</i> . When this value is <i>true</i> , any inter-site message that your site might generate will be transmitted to the destination site immediately (if possible). When this value is <i>false</i> , the message will be held until the remote site next initiates a “pull” operation.
SitMsgsPushPredelay	The minimum number of milliseconds <i>recipnetd</i> should wait after generating an inter-site message before transmitting it across the Internet. A nonzero value here increases the likelihood that a second message would be generated in the meantime, and thus that both messages could be sent in the same transmission for greater efficiency. There’s a tradeoff here between latency and traffic volume in inter-site replication. The recommended value is <i>2500</i> .
SitMsgsRecvDir	Complete directory specification (path) to the place where <i>recipnetd</i> should store message files for some of the inter-site messages it has received and processed. The directory cannot be used for any other purpose and may not contain extraneous files. In a default installation this would be <i>/var/recipnet/msgs-recv/</i> .
SitMsgsSentDir	Complete directory specification (path) to the place where <i>recipnetd</i> should store message files for all the inter-site messages it has transmitted. The directory cannot be used for any other purpose and may not contain extraneous files. In a default installation this would be <i>/var/recipnet/msgs-sent/</i> .
SitProviderCache	Specifies parameters for a cache in Site Manager that stores provider records that have been read from the database. For efficiency’s sake, the size of this cache should be equal to or greater than the total number of providers in the Reciprocal Net Site Network. The suggested parameters are <i>4096,0,32,0,0,0,false</i> . See RepHoldingIdsCache for a complete description of cache parameters.
SitRandomizerTask	Specifies parameters for a task in Site Manager that seeds the random number generator every now and then. The suggested parameters are <i>30000,2000,0</i> . See RepScannerTask for a complete description of periodic task parameters.
SitRedeliverHeldMsgsTask	Specifies parameters for a task in Site Manager that automatically redelivers “held” inter-site messages from other sites that encountered processing errors

previously. The suggested parameters are **2160000,120000,2**. See **RepScannerTask** for a complete description of periodic task parameters.

- SitReplayRequestTask** Specifies parameters for a task in Site Manager that “pulls” inter-site messages from other sites in the Site Network. The suggested parameters are **360000,120000,8**. See **RepScannerTask** for a complete description of periodic task parameters.
- SitRmiName** Similar to **RepRmiName**. In a default installation this value is **RecipnetSiteManager**.
- SitReceivedIsmReplayLimit** A performance-tuning option that defines the size of message batches exchanged during inter-site synchronization in some cases. The default value is **256**.
- SitSentIsmReplayLimit** A performance-tuning option that defines the size of message batches exchanged during inter-site synchronization in some cases. The default value is **256**.
- SitShunnedDuration** The duration (in milliseconds) that your site will avoid synchronizing with another site in the Site Network that is offline. After this timeout expires, synchronization may be attempted again. The default value is **14400000** (4 hours).
- SitSiteCache** Specifies parameters for a cache in Site Manager that stores site records that have been read from the database. For efficiency’s sake, the size of this cache should be equal to or greater than the total number of sites in the Reciprocal Net Site Network. The suggested parameters are **128,0,16,0,0,0,false**. See **RepHoldingIdsCache** for a complete description.
- SitStatsCommitTask** Specifies parameters for a task in Site Manager that “writes behind” in-memory statistics counters to ensure that usage statistics would not be lost if **recipnetd** were to crash suddenly. The suggested parameters are **360000,120000,11**. See **RepScannerTask** for a complete description of periodic task parameters.
- SitTidyUpTask** Specifies parameters for a task in Site Manager that releases memory resources and old pointers every now and then. The suggested parameters are **720000,30000,1**. See **RepScannerTask** for a



complete description of periodic task parameters.

SitUserCache	Specifies parameters for a cache in Site Manager that stores user records that have been read from the database. For efficiency's sake, the size of this cache should be equal to or greater than the number of user accounts on the local site. The suggested parameters are <i>256,0,16,0,0,0,true</i> . See RepHoldingIdsCache for a complete description of cache parameters.
SitWorkerThreadShutdownTime	Similar to RepWorkerThreadShutdownTime . This is <i>5000</i> in a default installation.
SitWorkerThreadStartupTime	Similar to RepWorkerThreadStartupTime . This is <i>5000</i> in a default installation.
SitWorkerThreadName	Similar to RepWorkerThreadName . In a default installation this value is <i>SiteManager</i> .

Configuration directives in web.xml

web.xml is a text file that lives in `/var/www/html/ recipnet/WEB-INF` , but most users find it convenient to edit the symlinked copy of the file in `/etc/ recipnet` . The file controls the operation of the web application, the portion of the Reciprocal Net site software that's hosted by the Apache Tomcat web application server. The contents of the text file are eXtensible Markup Language (XML) syntax. The root of the document is a `<web-app>` element. Beneath the root `<web-app>` element are a number of `<context-param>` elements. Each `<context-param>` element contains a single configuration directive; details about each of these may be found below. Other elements besides `<context-param>` are present below the root `<web-app>` element also, but these are not administrator-configurable and should be ignored.

Each `<context-param>` element, which contains a single configuration directive, is expressed in a syntax similar to the following:

```
<context-param>
  <param-name>hostname</param-name>
  <param-value>localhost</param-value>
  <description>
    The name of the host on which recipned is running
  </description>
</context-param>
```

. In this example, the text between `<param-name>` and `</param-name>` identifies this setting as the *hostname* directive. The description of the directive is present for this directive only as a convenience for the system administrator; it serves no functional purpose. The current value of the *hostname* directive is *localhost* .

The *web.xml* contains a number of configuration directives, each of which already has a value assigned. System administrators may change a directive's value, but should not alter its name or description. Syntax of the file should be preserved. Positioning of directives within the file is significant.

Currently the following configuration directives are defined in the file:

<code>adminEmail</code>	SMTP e-mail address of the server's administrator. In a default installation the value is blank, which implies that the administrator's e-mail address is <i>root@</i> your server's DNS name. It is not necessary to specify a value for this directive unless you wish to override the default behavior.
<code>childPath</code>	Specifies the <i>PATH</i> environment variable that should be used when by the web application when it launches external subprocesses. These programs might be spawned to perform specific scientific calculations or generate images, for example. The current release depends upon the programs <i>cjpeg</i> , <i>gs</i> , <i>pnmcrop</i> , <i>pnmdepth</i> , <i>pnmmargin</i> , <i>pnm scale</i> , <i>recipnet-ortep3</i> , <i>recipnet-renderclient</i> , <i>recipnet-vort2ppm</i> , and <i>sh</i> ; each of these programs must reside within the path specified. In a default installation this value is <i>PATH=/bin:/usr/bin:/usr/local/bin</i> .
<code>dartherServer</code>	The DNS name of a server offering molecular image rendering services (so-called "darther" services) to Reciprocal Net partner sites. At the time of this writing only once such server exists, the central rendering service hosted by IUMSC. The name of their server is <i>iumsc-beowulf.reciprocalnet.org</i> .
<code>defaultAuthenticatedApplet</code>	Controls which molecular visualization applet authenticated users should see by default when they first visit a <i>Sample detail</i> page. Possible values are <i>minijamm</i> , <i>jamm1</i> , and <i>jamm2</i> . The suggested value is <i>jamm2</i> .
<code>defaultAuthenticatedLevel</code>	The default "detail level" for samples to be used for authenticated users. This is not a security feature, because users are able to switch their level freely, but rather an educational one. The suggested value is <i>65536</i> , which is intended for use by "researchers". Another possible value is <i>8192</i> , which equates to "generic Common Molecules level".
<code>defaultUnauthenticatedApplet</code>	Similar to <code>defaultAuthenticatedApplet</code> above, except this setting controls the default molecular visualization applet for unauthenticated users. The



suggested value is *minijamm* because it is compatible with the widest range of web browser programs.

<code>defaultUnauthenticatedLevel</code>	Similar to <code>defaultAuthenticatedLevel</code> above, except this setting controls the default detail level for unauthenticated users. The suggested value is 65536 .
<code>externalProcessTimeout</code>	The maximum number of milliseconds an external program spawned by the web application (to perform a specific computation, for example) may run before it's assumed to have stalled and is forcibly terminated. The suggested value is 30000 (30 seconds).
<code>fileCacheTimeout</code>	The minimum number of seconds after having been generated that a molecular image or diagram is available for download by the user who requested it. Once this time has elapsed the image may no longer be available. The suggested value is 1800 seconds (30 minutes).
<code>hostname</code>	DNS or other name of the system on which <i>recipnetd</i> is running. It is possible to run <i>recipnetd</i> on one computer and the web application on potentially many other computers in order to improve performance. The suggested value <i>localhost</i> is correct for environments where <i>recipnetd</i> and the web application run on the same system.
<code>languageCacheParameterString</code>	These are performance tuning parameters used for a cache of localization resources. The default value is 64,0,4,0,0,0,true .
<code>languagePackJar</code>	The complete path of a jar file containing localization information for status codes, file codes, or action codes. The default value is blank, which means that Reciprocal Net site software's own language facilities are employed.
<code>maxFileCacheSize</code>	Specifies the maximum number of megabytes that temporary files generated in response to molecular diagramming or rendering requests may consume on the local filesystem. When this limit is approached, temporary files are deleted via a least-recently-used scheme. The suggested value is 512 megabytes.
<code>maxOaiPmhSamples</code>	The maximum number of samples to return in response to a single request from a digital library harvesting agent via the OAI-PMH protocol. Harvesting agents desiring more samples than this value are expected to send multiple consecutive

requests. Setting this value too high might expose your system to possible denial-of-service (DoS) attacks. Setting this value to *0* causes no limit to be enforced – all matching samples are returned in response to a single request – and poses an even greater DoS risk. The suggested value is *512*.

`newUserInitialPreferences`

The preference settings given to newly-created user accounts. The default value is *JaMM2,sample,detailed,3,10,true,false,true,false,true,false,true,false,false,form-based,false,false,false,false,true,true*

In order, the preferences are:

Visualization applet: valid choices are *minijamm*, *jamm1*, and *jamm2*.

Sample view: the preferred means of viewing sample records. Valid choices are *showsample* (for the “Sample Detail” page) and *sample* (for the “Edit Sample” page).

Sample detail: the preferred degree of detail to be shown on samples. Valid choices are *basic* and *detailed* .

Lab summary day count: the number of days to be displayed on the “Lab Summary” page.

Lab summary sample count: the number of samples to be listed in each box on the “Lab Summary” page.

Blank fields suppressed: whether blank fields are suppressed from display on the “Edit Sample” page. Valid choices are *true* and *false* .

Comments suppressed: whether workflow comments are suppressed from display on the “Edit Sample” page. Valid choices are *true* and *false* .

Empty file boxes suppressed: whether action boxes for file actions that are empty are suppressed from display on the “Edit Sample” page. Valid choices are *true* and *false* .

Empty correction boxes suppressed: whether action boxes for correction boxes that are empty are suppressed from display on the “Edit Sample” page.



Valid choices are *true* and *false* .

Empty action boxes suppressed: whether other action boxes that are empty are suppressed from display on the “Edit Sample” page. Valid choices are *true* and *false* .

Skipped-by-reversion action boxes suppressed: whether action boxes that were skipped by a reversion are suppressed from display on the “Edit Sample” page. Valid choices are *true* and *false* .

File descriptions suppressed: whether file descriptions are suppressed from display on the “Edit Sample” page. Valid choices are *true* and *false* .

Blank file listings suppressed: whether blank listings of files are suppressed from display on the “Edit Sample” page. Valid choices are *true* and *false* .

Suppression options suppressed: whether the suppression options box is suppressed from display on the “Edit Sample” page. Valid choices are *true* and *false* .

Files overwritten: whether newly-uploaded files can overwrite existing sample data files. Valid choices are *true* and *false* .

Upload mechanism: the preferred means of uploading sample data files. Valid choices are *form-based* and *drag-and-drop* .

Searches limited to non-retracted: whether searches are limited to samples that do not have a retracted status. Valid choices are *true* and *false* .

Searches limited to finished: whether searches are limited to samples that have a finished status. Valid choices are *true* and *false* .

Searches limited to local: whether searches are limited to samples with data files hosted at the local site. Valid choices are *true* and *false* .

Searches limited to provider: whether searches are limited to samples associated with the user’s affiliated provider. Valid choices are *true* and *false* .

Implicit preference changes: whether the user’s

preferences are updated implicitly as he navigates through the web application. Valid choices are *true* and *false* .

Space group symbol validation: whether space group symbols typed into the web application by the user should be validated. Valid choices are *true* and *false* .

<code>persistedOperationsCache</code>	These are performance tuning parameters used for a cache of persisted operations. The default value is <i>256,128,16,0,30000,.1,false</i> .
<code>repositoryManagerName</code>	The RMI registration name used by Repository Manager, one of the “core modules” contained within <i>recipnetd</i> . This value must match the <code>RepRmiName</code> value defined in <i>recipnetd.conf</i> . In a default installation it is <i>RecipnetRepositoryManager</i> .
<code>restrictLargeFilesAvailability</code>	Specifies whether prior versions of “large” sample data files are restricted to users affiliated with the sample, or whether such files are generally available. The sense of “large” is determined by the <code>RepPriorVersionFileSizeLimit</code> directive in <i>recipnetd.conf</i> . The suggested value is <i>true</i> because making available prior versions of very large sample data files consumes a great deal of server resources; thus, allowing unauthenticated users to do so might pose a security risk. If this value is <i>false</i> , then no file size limit is imposed upon unauthenticated users.
<code>rmiPort</code>	The TCP port number on which <i>recipnetd</i> is listening for connections from the site software web application. The value here must match the directive <code>GenRmiPort</code> in <i>recipnetd.conf</i> .
<code>shouldredirectAuthUserToSampleJsp</code>	Specifies whether authenticated users who execute a search that matches exactly one sample should be redirected to the <i>Sample Details</i> page for that sample (<i>false</i>) or to the <i>Edit Sample</i> page for that sample if they have such authorization (<i>true</i>). The default is <i>false</i> . Unauthenticated users are always redirected to the <i>Sample Details</i> page when a search they execute matches exactly one sample, regardless of this setting.
<code>siteManagerName</code>	Similar to <code>repositoryManagerName</code> above, except this is the name used by Site Manager. In a default installation this value is <i>RecipnetSiteManager</i> .
<code>stylesheetName</code>	The name of the CSS stylesheet that Reciprocal Net



pages should reference, as a path name relative to the webapp context root. The value should start with a slash (/) character. The default value is `/recipnet.css`.

`tempDir`

The filesystem location in which temporary files generated by the web application such as molecular diagrams and rendered images are stored. No more than `maxFileSizeCache` bytes will be stored in this directory at any one time. The default value is blank, which instructs the web application to store temporary files using the temporary file facility that Apache Tomcat provides. It is not necessary to specify a value for this directive unless you wish to override the default behavior.

`unauthenticatedPreferences`

The preference settings given to users who do not authenticate. The default value is *minijamm,sample,detailed,3,10,true,false,true,false,true,false,false,true,false,false,form-based,false,false,false,true,true*. See `newUserInitialPreferences` for a complete description of this value's syntax.

Format specification for .crt files

A *.crt* data file contains a representation of some part of a crystal structure, whether one or more particular molecules, the contents of a unit cell, or some other part of the structure as chosen by the file's creator. The contents describe atoms, the bonds between them, and optionally the unit cell dimensions and symmetry operations of the structure from which the data are drawn. Atom positions are expressed as Cartesian x, y, and z coordinates (in units of Ångstroms), hence the *.crt* extension. Symmetry and unit cell information is referred to the same Cartesian system. Among other uses, *.crt* files may be visualized using the JaMM family of applets included with the Reciprocal Net site software. The file format was developed by Indiana University in the 1990's; it is textual in nature, so *.crt* files can readily be manipulated via a standard editor.

Basic definitions

1. Each octet in the file must represent an ASCII character (in the range 0-127). Except as otherwise specified by this document, character codes 32 or less are prohibited.
2. A line break may consist of a carriage return character (ASCII code 13), a line feed (ASCII code 10), or a carriage return followed immediately by a line feed.
3. The characters space (ASCII code 32) and tab (ASCII code 9) are considered white space. Tokens are elements of the file separated by one or more consecutive white space characters. The precise number of white space characters separating two

- tokens is not significant. Additionally, a line break character may separate two tokens, optionally with white space on either side, when the specification calls for it.
4. The pound sign character (ASCII code 35, '#') signifies a comment. All characters from the pound sign to the next line break (or end of file, if there is no next line break) are ignored by parsers.
 5. A numeric value consists of a string of decimal digits (characters '0' through '9', ASCII codes 48 through 57), the decimal point (ASCII code 46), and the negative sign (ASCII code 45). If a negative sign is present then it must be the first character of the value. Real number values may contain at most one decimal point and are stored with at least the precision of the IEEE "single precision" standard format; this guarantees at least six significant decimal digits of precision. Integer values may not contain a decimal point and are stored with at least the precision of 32-bit two's-complement format.
 6. A textual value consists of between one and thirty-one sequential characters in the range 33-127, except that double quote (ASCII code 34), pound sign (ASCII code 35), and backslash (ASCII code 47) characters are not permitted.

Sections

The file is divided into sections, some of which are optional. Order of sections is significant, and those present **MUST** appear in the order they are listed below. At present, three sections are defined, but parsers **SHOULD** be tolerant of additional (unrecognized) sections that may be present at the bottom of a *.crt* file.

The CARTESIAN section

This section enumerates the atoms in the asymmetric unit of the structure, locates them in 3-space, and identifies any bonds between them. It is the only required section, and it must be first in the file. It is composed of three subsections:

1. A header line consisting of the word **CARTESIAN**, an integer count of the atoms that will be described by this section (subsection 2), an integer count of the bonds that will be described by this section (subsection 3), and a textual label for the file. Each value is separated from its neighbor by white space. The subsection is terminated by a line break. Atom and bond counts listed in this section **SHOULD** be considered advisory rather than prescriptive.
2. The ATOMS subsection contains zero or more atom definitions, one per line, starting on the line immediately after the one line of subsection 1. Each atom description consists of a textual label, a real number that represents the atom's x-coordinate, a real number that represents the atom's y-coordinate, a real number that represents the atom's z-coordinate, and an integer that represents the atomic number of the atom. Version 0.9.0 adds an additional, optional site code tag at the end of each atom line (see below). Each value is separated from its neighbor by white space. Any additional tokens on an atom line **SHOULD** be ignored by parsers. The order of atom descriptions



within this subsection is not externally significant, but the bond definitions in subsection 3 refer to this order. This subsection is terminated by a line starting with the word *ENDATOMS*; the remainder of this line SHOULD be ignored by parsers.

Site codes describe the relationship of atom records in the CRT file to crystallographic sites described in some external file, normally a CIF. They have the form of an atom site label, bar ('|', ASCII code 124) character, and CIF-style symmetry code (see the CIF dictionary) concatenated together, with any internal whitespace characters removed. Example: C10|2_455. Specific interpretation of these codes relies on reference to a corresponding external file, and unless the appropriate such file can be identified (normally by means of some local convention) it is safest to ignore these codes.

3. The BONDS subsection contains zero or more bond descriptions, one per line, each connecting two atoms in the model as identified by 1-based indices into the atom list of subsection 2. Bond descriptions consist simply of the two integer indices, separated by white space. Parsers SHOULD ignore the remainder of each line. Order of atom indices within a bond description and the order of bond descriptions within the subsection are not significant. Duplicate bond descriptions (in the same or the reverse order of atom indices) are permitted but not meaningful; they SHOULD be accommodated by parsers but not expected. This subsection is terminated by a line starting with the word *ENDBONDS*; the remainder of this line SHOULD be ignored by parsers.

The CELL section

This optional section describes the origin and unit cell vectors of the structure in terms of the Cartesian system to which the coordinates of the atoms in the *CARTESIAN* section are referred. This section is recognized by the keyword *CELL* as the first word of a line; the remainder of this line SHOULD be ignored by parsers. The rest of the section consists of four lines, each containing three real numbers constituting the coordinates of a point or vector in Cartesian space, separated by white space: the origin of the crystallographic coordinate system on the first line, then the \vec{a} , \vec{b} , and \vec{c} vectors, respectively, on each of the other three lines. Parsers MAY require that the last four lines of this section be devoid of other tokens. There is no explicit terminator for this section; the end is implicit in the fixed number of lines the section contains.

The SYMMETRY section

This optional section describes symmetry operations of the structure from which the model described in this CRT file is drawn, referred to the same Cartesian system in which the atomic coordinates are expressed. The identity operation is implied and MUST NOT be included in this section. This section is identified by the keyword *SYMMETRY* as the first word of a line. An integer count of the symmetry operations that will be described by this section follows the keyword on the same line, separated by white space; this count SHOULD be treated as advisory, not prescriptive. The remainder of the line SHOULD be ignored by parsers.

The body of the section consists of zero or more symmetry operation descriptions, each comprising four lines of three whitespace-delimited real numbers. The first three lines of each description contain the elements of the three rows of a 3-by-3 rotation matrix M (left-to-right, top-to-bottom), and the last row contains the elements of a translation vector \vec{t} . The transformation represented by M and \vec{t} , as applied to a Cartesian coordinate triple expressed as column vector \vec{p} to produce a Cartesian coordinate triple \vec{q} , is represented by the equation $\vec{q} = \vec{p} \times M + \vec{t}$.

The section is terminated by a line starting with the word **ENDSYMM**; the remainder of this line SHOULD be ignored by parsers.



Miscellaneous

Technical support

Complimentary technical support is available to all partner sites in the Reciprocal Net Site Network for as long as they remain active in the network. Additionally, pre-installation consulting is available to any crystallography lab considering joining the Site Network. Send e-mail to help@reciprocalnet.org. Telephone support also is available for urgent cases.

Technical support is provided by the Molecular Structure Center at Indiana University - Bloomington, the same people who wrote this release of the site software. Office hours are 8:00 am to 5:00 Eastern Time, Monday through Friday, excluding holidays.

Development team

This release of the Reciprocal Net site software was written by:

- John Bollinger, Ph.D.,
- Michael Durbin,
- Eguono “Iggy” Isiorho,
- Eric Koperda, and
- Christopher “Kit” Westneat.

The Reciprocal Net site software also contains previously-released contributions from:

- Amey Dharurkar,
- Ding “Dee” Feng,
- John Hanna,
- Anumit Jooloor, and
- Ryan Lauer,
- Yi “Richard” Li,
- Hsiu-chuan “Jane” Lin,
- Lei Qian,
- Nihar Sanghvi,
- Nihar Sheth, and
- Poornima Venkatakrishnan.

Documentation by Eric Koperda and John Bollinger.

Other contributors

General project guidance by John Huffman, Ph.D., John Bollinger, Ph.D., Gary Wiggins, Ph.D., William Harwood, Ph.D., and Gerry Bernbom, Ph.D.

Functional advice by Maren Pink, Ph.D.

Other contributing investigations by Jon Dunn and Randall Bramley, Ph.D.



Acknowledgements

Partial support for the Reciprocal Net Project in the years 2001 through 2005 was received from the U.S. National Science Foundation via grant DUE-0121699.

Partial support for the Reciprocal Net Project in the years 2007 and 2009 was received from the U.S. National Science Foundation via grant IIS-0513768.

Continued support for the Reciprocal Net project in the years 2001 through 2009 was received from various units within Indiana University.

The *recipnet-webapp* RPM package includes a binary, modified version of ORTEP-III, the Oak Ridge Thermal Ellipsoid Plot Program written by Carroll K. Johnson and Michael N. Burnett of the Oak Ridge National Laboratory (USA). This public domain software is redistributed in accordance with its standard license by the Reciprocal Net project. Please visit <http://www.ornl.gov/ortep/> for more information.

The *recipnet-site-webapp* RPM package includes binary, modified versions of programs from the VORT package, a very ordinary rendering toolkit. The software is in the public domain and is redistributed in accordance with its standard license by the Reciprocal Net project. Please visit <http://www.wumpus.com.au/eric/> for more information.

The *recipnet-site-utils* RPM package includes a binary version of SHORTEP, a utility for converting crystallographic data files from one format to another. The software was created by the Indiana University Molecular Structure Center (IUMSC) and is maintained as part of their XTEL collection of crystallography programs. SHORTEP borrows code heavily from ORTEP-II created by Oak Ridge National Laboratory. SHORTEP is Copyright ©2002, Trustees of Indiana University and is redistributed by the Reciprocal Net project through express permission from IUMSC. Please visit <http://www.iumsc.indiana.edu/> for more information.

Apache, Apache Httpd, and Apache Tomcat are trademarks of The Apache Software Foundation, and are used with permission.

MySQL is a registered trademark of MySQL Inc.

Red Hat, Red Hat Linux, and Red Hat Network are registered trademarks of Red Hat, Inc.

Sun, Java, JavaServer Pages, and J2EE are registered trademarks of Sun Microsystems, Inc.

Microsoft, Windows, and Internet Explorer are registered trademarks of Microsoft Corporation.

Apple, Macintosh, MacOS, and Macintosh Runtime for Java are registered trademarks of Apple Computer, Inc.

Mozilla is a registered trademark of The Mozilla Organization.

GNU is a registered trademark of the Free Software Foundation.

License

The Reciprocal Net site software described by this document was created by the Reciprocal Net project. Copyright © 2002-2009, The Trustees of Indiana University.

Reciprocal Net site software is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

Reciprocal Net site software is distributed 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 (<http://www.gnu.org/licenses/>) for more details.

In addition to the license described above, the Trustees of Indiana University may distribute the Reciprocal Net site software under other terms.