

Helgrim.com

Navigation

[\[Index of articles\]](#)

[Home](#)
[About me](#)
[Contact](#)

Software
FreeBSD
[Fujitsu-Siemens C1020](#)
[6.2 Install](#)
Sendmail
[Stress testing](#)
PostgreSQL
[Altering database schemata](#)
Documentation
[DBMail](#)
[Request Tracker](#)
Links
[Miscellaneous](#)

Advertising



Content

[1 Requirements](#)

- [1.1 Perl](#)
- [1.2 Database backend](#)
- [1.3 Perl enabled webserver.](#)

[2 Unpack.](#)

[3 Configure the Makefile.](#)

[4 Check dependencies](#)

- [4.1 Building Apache with mod_perl](#)

[5 Install](#)

- [5.1 Upgrading from RT 2.0.x](#)
- [5.2 Upgrading from RT 1.0.x](#)

[6 Final configuration](#)

[7 Configuration](#)

- [7.1 Configuring the email gateway](#)
 - ◆ [7.1.1 exim](#)
 - ◆ [7.1.2 Qmail](#)
- [7.2 Configuring the web interface](#)
 - ◆ [7.2.1 Apache mod_perl](#)
 - ◆ [7.2.2 using a seperate webserver with mod_proxyypass](#)
 - ◆ [7.2.3 speedycgi Configuration](#)
 - ◆ [7.2.4 fastcgi Configuration](#)

[8 Users and Queues and Scrips. Oh my!](#)

[9 Appendix](#)

- [9.1 Installation on Debian Linux](#)
-

1 Requirements

You should have downloaded the latest release of RT. [\[www.fsck.com/projects/rt/download\]](http://www.fsck.com/projects/rt/download)

1.1 Perl

Perl5.005_03 or later with support for setgid perl scripts [<http://www.perl.com>]

RT's command line and mail gateway tools run setgid to the 'rt' group to protect perl setup to support "setuid scripts".

RT also relies over 30 perl modules to be installed. A tool is included to automa

IM Status



1.2 Database backend

RT requires one of the following database backends:

MySql v3.23.38 or higher. [www.mysql.org]

RT is primarily developed on MySql. Because of this, it's likely to be the best-te

PostgreSQL 7.1.1 or newer. [www.postgresql.org]

Oracle [www.oracle.com]

RT runs on Oracle 8 or 8i, though there are currently known issues with Oracle

1.3 Perl enabled webserver.

Apache 1.3 [httpd.apache.org]

mod_perl [perl.apache.org]

In order for the web interface to work, apache must be compiled with mod_perl

[\[Table of Contents\]](#) [\[Top of section\]](#)

2 Unpack.

Unpack the distribution somewhere other than where you want to install RT. To do this

```
% tar zxvf rt.tar.gz -C /tmp
```

[\[Table of Contents\]](#) [\[Top of section\]](#)

3 Configure the Makefile.

Check through `/tmp/rt/Makefile`

There are many variables you NEED to customize for your site. Even if you are just upg

Here are the some of the more likely candidates to need changing:

RT_PATH - Where you want the installation to live.
`/opt/rt2` is default, `/usr/local/rt2` may be more to your taste.

RT_MAN_PATH - Where to install the man pages for RT.
You may wish to change to `/usr/share/man` or `/usr/local/man`

RT_LOG_PATH - Where to write log files.
RT writes numerous log files based on pid's, so you may want to make a directory just f

DB_TYPE - Which database backend to use.
Choices are:

- Pg for PostgreSQL;
- mysql for Mysql;
- Oracle for, well, Oracle.

DB_HOME - Where to find the executables and libraries for your database
`$DB_HOME/bin` should contain the binaries themselves - e.g. if

```
$ which mysql
```

gives `"/usr/local/mysql/bin/mysql"`, `$DB_HOME` should be `"/usr/local/mysql"`

DB_DBA - A user who has permission to create new databases and database users.
This might be "root" for Mysql, "postgres" or "pgsql" for PostgreSQL, or "system" fo

DB_RT_USER - A user who will actually use the database
If the username does not already exist, it will be created.

WEB_USER and WEB_GROUP - The user that your webserver runs as. This shou
configuration file for the User directive.

[\[Table of Contents\]](#) [\[Top of section\]](#)

4 Check dependencies

Satisfy RT's myriad dependencies:

The Makefile has two targets, *testdeps* and *fixdeps*, which automate this. Make sure y

```
% cd /tmp/rt
% make testdeps
```

This will check that all the perl modules which RT depends on are installed. If there are unsatisfied dependencies, install them by hand or run

```
% make fixdeps
```

You may need to install some modules by hand, especially Apache::Session, Apache

```
% perl -MCPAN -e 'install DBD::mysql::Install'
```

Alternatively, you should be able to find all the necessary modules at <http://www.he>

Now make sure everything is installed:

```
make testdeps
```

Repeat until happy.

4.1 Building Apache with mod_perl

The mod_perl guide can be found at: [<http://perl.apache.org/guide>]

The following is an excerpt, which should work for most people. Download t

```
% tar xzvf apache_x.x.x.tar.gz
% tar xzvf mod_perl-x.xx.tar.gz
% cd mod_perl-x.xx
% perl Makefile.PL APACHE_SRC=../apache_x.x.x/src USE_APACI=1
% make && make test && make install
% cd ../apache_x.x.x
% make install
```

This builds Apache statically with mod_perl, installs Apache under the default /usr/lo

[\[Table of Contents\]](#) [\[Top of section\]](#)

5 Install

If this is a new installation, follow the steps below. If you are upgrading from an earlier version, see the upgrade instructions.

1. Create a group called 'rt'

2. As root, type:

```
# make install
```

This will install RT, and configure your database. If it fails, you must clear the database and re-run the installation.

```
# make dropdb
```

5.1 Upgrading from RT 2.0.x

If you are upgrading from rt.2.0.x:

1. Back up your existing /rt2/etc/config.pm The installation will save your existing configuration file.

2. Rebuild the libraries as root:

```
# make upgrade
```

This will build new RT wrappers, files and libraries without overwriting your RT configuration. You should copy your old configuration file into its place.

5.2 Upgrading from RT 1.0.x

Request Tracker went through major changes from version 1 to version 2. The upgrade tool will help you upgrade from version 1 to version 2.

The import tool will import the following information from your rt1 database:

- Queues, Areas, Users, Acls, Mailing Rules, Queue Members, Tickets and Transitions

It will not, however, import:

- Attachments which were removed by stripmime, or Templates.

Be warned, if you have a sizeable installation, the import can take some hours, so be patient.

1. Make sure you have completed the previous steps for installing RT2. It is important that you have the correct version of RT2 installed.

2. Cd to your rt1 installation.

Helgrim.com - Feargal Reilly

```
$ cd /path/to/rt/etc
```

3. Download import-1.0-to-2.0 from [\ftp://ftp.fsck.com/pub/rt/contrib/2.0/rt-ad

4. If your RT 1.0 transaction history doesn't live on the host you're running the i

5. Edit the configuration defaults in import-1.0-to-2.0

Set \$DEFAULTQUEUE to the name of one of your RT 1.0 queues.

If you do not do this, THE IMPORT WILL FAIL. Also, make sure the Minimum does, your users will NOT be imported.

6. If you are importing to a postgres database, you will need to execute the follow

```
$ psql rt2
psql> SELECT setval('tickets_id_seq', (select max(id) from ticket
```

7. Do the import by typing:

```
$ perl ./import-1.0-to-2.0
```

The script will run, without requiring user intervention. If you are importing a large number of tickets, it may take some time. If you are importing a large number of tickets, they will have been imported correctly.

[\[Table of Contents\]](#) [\[Top of section\]](#)

6 Final configuration

Edit etc/config.pm in your RT installation directory. In many cases sensible defaults have been included. In others, you MUST supply a value.

Configure the [email](#) and [web](#) gateways, as described below.

Stop and start your webserver, so it picks up your configuration changes.

N.B. Remember to do this any time you make changes to config.pm

Using rtadmin, or the web interface add a RT user for yourself.

This must be done as root, as no other user has permission to do any of these things. Right now, root's password is 'password'. **You should change that now.**

Commandline instructions:

```
[root@host bin]# ./rtadmin --user --create --password=""
```

Web instructions:

Log into the web ui as root

Click on "Administration"

Click on "Users"

Click on "Create a new user"

NOTE: root's password for the web interface is "password"

(without the quotes.) Not changing this is a SECURITY risk

[\[Table of Contents\]](#) [\[Top of section\]](#)

7 Configuration

7.1 Configuring the email gateway

An alias for the initial queue will need to be made in either your global mail aliases file (if you are using NIS) or locally on your machine.

Add the following lines to /etc/aliases (or your local equivalent)

```
rt-comment: "|/path/to/rt2/bin/rt-mailgate --queue general --action  
rt:         "|/path/to/rt2/bin/rt-mailgate --queue general --action  
           |  
           ----/  
           |  
           ---/
```

7.1.1 exim

```
From: seph  
Date: 10 Aug 2001 18:07:09 -0700  
User-Agent: Gnus/5.0803 (Gnus v5.8.3) Emacs/20.7  
Subject: [rt-users] rt-mailgate and exim  
List-Id: For users of RT: Request Tracker
```

I recently installed rt2 onto a machine that was using exim as it's MTA. I hacked
exim.conf for a bit (with help from a friend) and ended up with a machine that

basically I told exim that I had a local user "rt" (I don't) and that - was a p
just execs rt-mailgate with the relevant command line flags.

my end setup will have my hacked exim.conf on the rt2 host, and my de

```
abuse: rt-abuse+correspond@rt2
```

I find this much simpler than having big long aliases somewhere. anyho

in the TRANSPORTS section of my exim.conf I've added:

```
# my rt hack
local_delivery_rt:
  driver = pipe
  command = "/usr/local/rt2/bin/rt-mailgate --queue ${quot
  return_path_add
  return_output
  prefix = ""
  suffix = ""
  user = www-data
```

and in the DIRECTORS section I've added:

```
localuser_rt:
  driver = smartuser
  suffix = +*
  prefix = *-
  new_address = ${quote:${lc:${local_part}}}@${domain}
  transport = local_delivery_rt
```

7.1.2 Qmail

rev 1.0 - 25 June 2001 - Karel P Kerezman

This short document will attempt to deal with the issue of how to make much detail about how queues are created and administered in RT2, no toys, sonny, you're old enough to read the appropriate instruction manu

If you're migrating from the original RT to RT2, you may face a bit of a RT2 administrators as well. Here we sum up the basic set of problems:

- ◇ The usual place to put aliases in Qmail is ~alias. These aliases r mine could probably find a way to make this work, but I found
- ◇ The solution I've come up with will probably require a change yours. You've been warned.

And so, without further ado, here's what you should do if you want to b

1. For each queue, create two files in ~rt, a .qmail-queue and .qmail-officecomment that contain the following, respectively:

```
|/opt/rt2/bin/rt-mailgate --queue Office --action co
|/opt/rt2/bin/rt-mailgate --queue Office --action co
```

2. Set the correspond and comment addresses for your queue to rt-office@lancelot.kgon.com and rt-officecomment@lancelot.kgon.com. You may want to come up with a better way to do it now that you've seen how

That's really all there is to it. But I hear you asking, "Why do I have to do this?" Qmail handles aliases, and the way permissions affect the operation of mail (e.g. /var/qmail/alias) and if the alias is a piped system command that command will not quite cooperate when it comes to running RT2's suid-wrapped mailgate.

There is one user that can run rt-mailgate with impunity, though, and that is root. Of course, the only way these aliases will be called is if email is sent to them.

A quick tip: If you're like me, and too few are, you want to have an "incoming" General queue, and we have an easy-to-remember email address available for that.

1. You might as well have a "plain" .qmail file in order to deal with incoming mail.

```
|/opt/rt2/bin/rt-mailgate --queue General --action incoming
```

2. Then make sure that the General queue is set appropriately to action incoming.
3. Use rt@hostname as the recipient in any single-page web form. (I'll be adding users along with the basic run-down on how RT2 will work for you. See <http://washuu.kgon.com/addto.htm>.)

This concludes my Qmail and RT2 documentation. Comments, suggestions, and corrections to our pet betta fish. Future revisions may appear pending inspiration and time.

7.2 Configuring the web interface

Note, that after applying changes to the configuration file or the perl library, the perl command will NOT work, as Mason's cache will not be cleared. You must restart the web server.

RT's web ui is based around HTML::Mason, which works best with the mod_perl. mod_perl is apache's your best bet.

7.2.1 Apache mod_perl

RT Uses HTML::Mason. You'll need to add a few lines to your httpd.conf. (If you're not faster around with it), you can do something like this:

```
<VirtualHost your.ip.address>
DocumentRoot /path/to/rt2/WebRT/html
ServerName your.rt.server.hostname
PerlModule Apache::DBI
PerlFreshRestart On
PerlRequire /path/to/rt2/bin/webmux.pl
<Location />
    SetHandler perl-script
    PerlHandler RT::Mason
```


7.2.3 speedycgi Configuration

The speedycgists on version of RT's web interface isn't supported in RT

7.2.4 fastcgi Configuration

Configure your web server to support fastcgi scripts. Tell your web serv

```
ScriptAlias / /opt/rt2/bin/mason_handler.fcgi/
```

[\[Table of Contents\]](#) [\[Top of section\]](#)

8 Users and Queues and Scrips, Oh my!

Run through the installation

Create your users

In its default configuration, RT uses an internal users database to keep track of who can access RT and who has what rights within the system.

One of your first tasks should be to create users for anyone who will need to work with tickets within RT.

With the web interface, click on "Config" then "Users" then "Add new user". When creating new users, be sure to fill in:

Name

Email Address

Password

Be sure to click on "Let this user to be granted rights"

Create your queues

RT's primary administrative unit is the 'queue.' Use queues to separate tickets either by the groups of people who should be dealing with the

Helgrim.com - Feargal Reilly

the user community or ticket attributes, or some combination therein. Generally, you don't want to create queues for time-limited projects. In most situations, queues are not things you should be creating or deleting with any frequency.

Grant the users the rights they need.

RT provides a rich access control system that allows rights to be granted to groups, individual users and users in specific roles.

To allow arbitrary remote users to submit tickets into a given queue via email, grant "Everyone" the rights to "SeeQueue", "CreateTicket", "ReplyToTicket" and "CommentOnTicket" for that queue.

If you intend to let ticket requestors use the requestor-mode web interface, you should grant "Requestor" the right to "ShowTicket"

To make sure that your staff can work with tickets, you should grant them all the following additional rights:

```
"ShowTicket"  
"ShowTicketComments"  
"Watch"  
"WatchAsAdminCc"  
"OwnTicket"  
"ModifyTicket"
```

Set up Scripts

RT's "Scripts" system lets sites configure what email gets sent by RT in response to ticket creation or updates. Unless you set up Scripts, RT will not send any email. (It's also a general extension mechanism which allows changes to persist across updates. But if you're reading this quickstart guide, you probably just don't want to go there yet.)

Scripts are composed of a Condition, an Action and a Template. A number of Conditions and Actions come with RT. Others are available as contributed add-ins from members of the RT community. Later on, you can define custom templates for each queue, but for now, RT ships with a set of predefined templates which should get you started.

A reasonable default set of scripts for a queue is something like this:

```
OnCreate AutoReplyToRequestors with Global Template: AutoReply  
OnCreate NotifyAdminCcs with Global Template: Transaction  
OnCorrespond NotifyAllWatchers with Global Template: Correspondence  
OnComment NotifyAdminCcsAsComment with Global Template: AdminComments  
OnResolve NotifyRequestors with Global Template: Resolved
```

Set up queue watchers

RT lets you set up Cc and Administrative Cc watchers globally and for each queue. (Note that mail isn't sent unless you define Scripts to send it). Generally, Cc watchers are folks who get notifications of ticket creation, updates and can view the queue (though this, of course, depends on how you configure things). Administrative Ccs are usually the folks who have the rights to modify tickets. Set up folks you want to get Cc or Administrative Cc mail as watchers for your queue in "Configuration"/"Queues"/"Queue / Watchers".

Set up keywords

RT's keyword system is centered around a single global hierarchy of

Helgrim.com - Feargal Reilly

keywords and the concept of "Keyword Selections". Keyword Selections allow you to set up custom fields either for your whole RT installation or for a single queue. Keyword Selections let you pick a node in the Keywords tree as its "root" and to decide whether users can pick either a single value or multiple values from that list.

For now, you probably don't need any keyword selections, but once you want to start 'tagging' requests with keywords, come back and add a keyword called "Queues" to the Keywords hierarchy ("Configuration"/"Keywords").

Once you've created that keyword, click on it and add a new keyword named after your queue. Click on that. Add a keyword as the root of your new Keyword Select. Let's call it "operating system." Under Operating system, add a bunch of operating systems.

Now, click on "Configuration" / "Queues" / your queue name / "Keyword Selections" Enter a name for your keyword selection, like "Client OS". Allow "multiple" values of "Queues//Operating System Up to 0 (unlimited) levels deep.

Now you'll be able to set 'Client OS' for tickets in your queue.

[\[Table of Contents\]](#) [\[Top of section\]](#)

9 Appendix

9.1 Installation on Debian Linux

```
Subject: installing rt2 onto debian
From: seph
Date: 09 Aug 2001 17:17:06 -0700
User-Agent: Gnus/5.0803 (Gnus v5.8.3) Emacs/20.7
```

okay, my list of steps, including what packages I needed. there of course, a bunch of post install stuff, but it's got the packages needed.

seph

```
# get source
cd /usr/local/src
wget http://www.fsck.com/pub/rt/release/rt.tar.gz
tar xzf rt.tar.gz
```

```
#edit the makefile
```

Helgrim.com - Feargal Reilly

```
# test deps:
make testdeps

# install needed debs
# NOTE: debian's perl packaging is fucked. I had to build many of
#       myself so they'd work with the version of perl that's ins
#       apt-get source -b

perl-suid

libapache-mod-perl

libapache-dbi-perl
libhtml-parser-perl
libparams-validate-perl
libhtml-mason-perl
libmime-perl
libfreezethaw-perl
libapache-request-perl
libapache-session-perl
liblogfile-rotate-perl
libtext-template-perl

# something in debian should depend on this but doesn't
libmd5-perl

# package and install more perl modules.
# yay cpan->dpkg
liblog-dispatch-perl_1.79-1_all.deb
libdbix-datasource-perl_0.02-1_all.deb
libdbix-searchbuilder-perl_0.41-1_all.deb
libtext-wrapper-perl_1.000-1_all.deb
libtie-ixhash-perl_1.21-1_all.deb

# create the rt group, and add www-data to it.
addgroup rt          adduser www-data rt

# now install (we have a db, so make upgrade)
make upgrade
```

[\[Table of Contents\]](#) [\[Top of section\]](#)



© Feargal Reilly 2003-2010 | feargal@helgrim.com | [Hosting by FBI](#) | [Valid HTML 4.01](#) | [Valid CSS](#)