

Chapter 1: Introduction	
About ticketing .....	4
About RT.....	4
RT concepts.....	5
<b>Chapter 2: Installing RT</b>	
<b>Requirements .....</b>	<b>8</b>
RT .....	8
Perl.....	8
Perl modules .....	8
Database backend .....	8
Webserver.....	9
<b>General Installation .....</b>	<b>9</b>
<b>Setting Up The Web Interface .....</b>	<b>14</b>
Configuring Apache .....	14
<b>Setting up the mail gateway.....</b>	<b>15</b>
<b>Configure RT.....</b>	<b>16</b>
Create users.....	16
Grant user rights.....	16
Create queues .....	17
Set up scrips .....	17
<b>Chapter 3: Using RT on the Web</b>	
<b>Logging in.....</b>	<b>19</b>
<b>Home page interface.....</b>	<b>19</b>
RT at a glance .....	20
Top navigation bar .....	20
Left side navigation bar .....	20
<b>Tickets interface.....</b>	<b>20</b>
Display .....	21
History .....	21
Basics.....	22
Dates .....	22
People .....	22
Relationships/Links .....	24
Jumbo .....	25
<b>Common tasks.....</b>	<b>25</b>
Searching.....	25
Creating a new ticket .....	26
Replying and commenting .....	27
Reply and comment integration with RTFM.....	27
Modifying a ticket .....	28
Modifying people.....	29
Modifying relationships .....	29
Updating many tickets at once.....	29
Resolving a ticket .....	29
<b>Changing your own user information.....</b>	<b>30</b>
Personal groups .....	30
Delegating.....	30
<b>Logging out .....</b>	<b>31</b>
Security.....	33
Rights .....	34
Custom fields .....	34

Scripts.....	34
Approval.....	35
<b>Chapter 4: Adminning RT</b>	
<b>Users.....</b>	<b>35</b>
Creating a user .....	35
Searching for a user.....	36
Modifying a user .....	36
Granting user rights.....	36
Revoking user rights .....	37
Disabling a user.....	37
Re-enabling a user.....	37
<b>Groups.....</b>	<b>38</b>
Creating a group .....	38
Changing a group's name or description .....	38
Adding group members .....	38
Removing group members.....	38
Granting group rights.....	39
Revoking group rights .....	39
<b>Queues.....</b>	<b>40</b>
Creating a queue .....	40
Setting priority.....	40
Finding a queue.....	41
Modifying a queue.....	41
Adding watchers to a queue.....	41
Deleting watchers from a queue .....	41
Scripts in queues .....	42
Creating a scrip.....	42
Modifying a scrip.....	42
Deleting a scrip .....	42
Templates in queues .....	42
Creating a new scrip template.....	42
Modifying a scrip template .....	42
Deleting a scrip template .....	43
Custom fields in queues .....	43
Adding a custom field.....	43
Modifying a custom field .....	43
Disabling a custom field.....	43
Re-enabling a custom field .....	43
Rights within a queue.....	43
Revoking queue rights.....	44
Disabling a queue .....	44
Re-enabling a queue.....	44
<b>Global.....</b>	<b>44</b>
Global scrips .....	44
Adding a scrip .....	44
Modifying a scrip.....	44
Deleting a scrip.....	45
Global templates.....	45
Creating a scrip template.....	45
Modifying a scrip template .....	45
Deleting a scrip template .....	45
Global custom fields .....	45
Global group Rights.....	46
Global user Rights .....	46

<b>Appendix 1: Glossary .....</b>	<b>48</b>
<b>Appendix 2: Rights .....</b>	<b>53</b>
<b>Appendix 3: Scripts .....</b>	<b>56</b>
<b>Appendix 4: Building Apache with mod_perl</b>	<b>60</b>
<b>Appendix 5: Creating Approvals</b>	<b>62</b>
Appendix 6: Perl Modules .....	67

# Chapter 1

## Introduction

Welcome to RT, your new favorite ticketing system. This introductory chapter will explain what ticketing is and what RT is. If you're already familiar with RT, you can skip ahead to another chapter.

### About ticketing

When most people say "ticketing," they're talking about buying a stub of paper that gets them into an event. Our type of ticketing is a way for organizations to keep track of their to-do lists, who is working on which tasks, what's already been done, and when tasks were completed. You will also hear the phrase "trouble ticketing" bandied about, though there does not have to be a problem for ticketing to be useful.

Ticketing can make life easier for many people and departments in an organization -- basically anyone who needs to keep track of a list of tasks. And it can be used with a variety of tasks, like fixing software bugs and answering customer service requests.

Since tickets are detailed records, you might not want to bother using it for one-off events ("Buy Bob in HR a going-away present.") You also shouldn't use it for tasks that are too broad ("Save world") or too granular ("Go to phone." "Pick up phone" "Lift finger." "Dial 9.")

### About RT

RT is a ticketing system for small- to medium-sized organizations. It is available under the terms of the GNU General Public License (GPL), so it doesn't cost anything to set up and use. We'll get into the technical features later, but for now, let's look at how RT might work:

- Billy has a problem with his Bananamaster 3000, so he sends email to help@example.com
- RT records the new issue in its database and sends Billy an electronic "ticket stub" containing the new ticket's ID number and, ideally, some helpful information. He can refer to the ticket stub if he needs to send another email later.
- RT forwards Billy's email to a set of staff members.
- One of the staff members, Jane, is the company's Bananamaster expert. So Jane takes over Billy's issue and drafts a reply to Billy, which RT automatically records and forwards to him.
- Billy is a happy customer; Jane resolves the issue and heads down to the pub for a pint.

This workflow is flexible, can be customized for your organization, and need not involve bananas.

## RT concepts

There are some words and phrases you should understand in the context of RT before you begin. See *Appendix 1: Glossary* for more definitions.

A **ticket** is the central object in RT, the thing that needs to get done. Tickets have metadata attached to them such as an owner, status, and queue -- we'll get to all that in a minute.

A **queue** is the central administrative domain of RT; it keeps things organized. As the name implies, it's a line of tickets waiting to be worked on, but it's also, to some extent, the ticket's category. For instance, you might have the right to create, delete, and comment on tickets in the Foo queue, but only the right to comment on tickets in the Bar queue.

A ticket's **history** is what it sounds like: everything that's happened to a ticket. Facets of ticket history could include when the ticket was created, how it has changed, and any comments about it or replies to it. Like real history, ticket history cannot be changed (at least not without messing up the space-time continuum), so be aware that any comments you make about a ticket are permanent.

**Ticket updates** can take one of two forms. A **reply** is a public remark that a requestor can see. A **comment** is a private note for staff not visible to the requestor. This is useful when you want to be tactful but still convey important information, like, "This requestor is an investor, so be nice" or "This user's request mentioned his 'PC' but he really has a Mac."

**Priority**, how important a ticket is, is represented as numerical scale from 0-99, with 99 being the highest priority. By setting a **final priority**, you can make a ticket's priority increase or decrease as its **due date** draws closer. The difference between 20, 50, and 75 may vary from organization to organization, but there should be an organizational policy or every ticket will get rated 99 by anxious users. *Chapter 4: Adminning RT* contains a useful model for setting organizational policy for priority.

The **status** drop-down menu offers several choices for classifying each ticket. Here's what they mean

**new:** the ticket has just been created and hasn't been touched yet.

**open:** the ticket is getting worked on

**stalled:** due to circumstances beyond your control (waiting for the requestor to respond, waiting for the owner to return from Sri Lanka), the ticket isn't getting worked on right now. It will open again when someone adds a comment or reply.

**resolved:** hooray! Work on the ticket has been completed and is out of everyone's hair.

**rejected:** the ticket is not the staff's problem and is not going to be resolved, but is, for some reason, worth recording in the system. For instance, if an employee asks approval for something ridiculous, you can reject the ticket, but it will stay in the database as evidence that the employee makes silly requests.

**deleted:** the ticket never should have been in the system -- it was spam, it was a list of passwords, it was porn, use your imagination -- and is now being zapped for good.

A **watcher** is someone who is interested in a ticket. You'll find these filed under "People" when you're looking at a ticket. There are several types, or **roles**, of watchers:

**owner:** the person responsible for the ticket and its resolution. Each ticket can have only one owner.

**requestor:** the person or people who asked for something to get done; the ticket's reason for being.

**Cc:** someone (or someones) who should get copies of any replies that go to the requestor. This might be the requestor's boss, sales rep, etc. This person will see the email but doesn't necessarily have the right to work on the ticket.

**Admin Cc:** a Cc or Ccs that also gets copies of comments and generally are allowed to work on the ticket.

We're hedging with words like "generally" and "necessarily" because any role can be assigned any right

**Relationships** in RT can be ticket-to-ticket, but can also link tickets to external items like URLs or FedEx shipping numbers. For the sake of simplicity, we'll refer only to tickets in the following explanation of relationship types:

**depends on:** the ticket can't be resolved unless another ticket is also resolved. The converse is **depended on by**.

**refers to:** the ticket doesn't *need* the other ticket, but it would sure be useful for you to look at it. The

converse is **referred to by**.

**parent:** a big, general ticket ("Move house.")

**child:** a subproject of a parent ("Hire movers." "Pack." "Eat pizza.")

**Custom fields** are, yes, database fields that your organization can make up according to its needs. For details, see the *Custom fields* section in *Chapter 4: Adminning RT*.

**Scripts** are custom notifications that will automatically take Action X in response to Condition Y. You could have RT notify the requestor when a ticket is resolved, for instance, or have RT page you when your boss submits a request. For details, see the *Scripts* section in *Chapter 4: Adminning RT*. And no, we didn't misspell "scripts." We made up a new word. Try it, it's fun.

Now that you're clear on the concepts, you're ready to start using RT.

**DRAFT, April 13, 2003**

# Chapter 2

## Installing RT

### Requirements

Read this now to save yourself a million headaches later. To properly install RT, you'll need some things:

#### RT

The latest version is available at <http://www.bestpractical.com/rt/download.html>

#### Perl

If you will be running RT under FastCGI, you need support for setgid perl scripts.

---

Note: If you intend to use RT's support for **non-English languages**, we strongly recommend you use Perl 5.8.0 or newer, as it contains significantly enhanced support for Unicode and non-English languages. In other words, trying to use RT with Perl 5.6.1 in a non-English-speaking environment will hurt a lot.

---

#### Perl modules

If that sounds vague, don't worry: a tool included with RT takes care of the installation of most of these automatically during step 3 below. See *Appendix 6: Perl Modules* for a complete list of the required modules in RT 3.0.0.

The tool supplied with RT uses Perl's **CPAN** system (<http://www.cpan.org>) to install modules. Some operating systems package all or some of the modules required and you may be better off installing the modules that way.

#### Database backend

We recommend **MySQL** (<http://www.mysql.com>), because RT was primarily developed on MySQL, so it is the best-tested database for RT. We currently support **version 4.0.9 or newer** with support for InnoDB (some older releases had crippling SQL bugs).

**PostgreSQL 7.2 or later** will also work (<http://www.postgresql.org>)

## Webserver

We recommend using **Apache version 1.3.x or 2.x** (<http://httpd.apache.org>) compiled statically with **mod\_perl** (<http://perl.apache.org>) or a webserver with **FastCGI** support (<http://www.fastcgi.com>). We've had reports that 2.x does not work with mod\_perl2. mod\_perl 1.x must be build with EVERYTHING=1

For an excerpt, see *Appendix 4: Building Apache with mod\_perl*

Mod\_perl 2.0 isn't quite ready for prime time just yet. We strongly recommend that sites use Apache 1.3 or FastCGI.

Compiling mod\_perl on Apache 1.3.x as a DSO has been known to cause stability problems with HTML::Mason, which RT runs on top of, so we don't recommend it. See the web page [http://www.masonhq.com/docs/faq/#why\\_am\\_i\\_getting\\_segmentation\\_fa](http://www.masonhq.com/docs/faq/#why_am_i_getting_segmentation_fa) for more detail.

RT's FastCGI handler runs setgid to the "rt" group to protect RT's database password. You may need to install a special "suidperl" package or reconfigure your perl setup to support "setuid scripts" if you intend to use RT with FastCGI.

- **Debian GNU/\* 3.0+**: the package which installs suidperl is called perl-suid, and should work without any tweaking.
- **FreeBSD 4.2+**: the package is called sperl, and should install a suidperl that just works
- **Conectiva Linux 6.0+**: suidperl is installed by default when perl is installed, but the program /bin/suidperl is not setuid. You must use *chmod* to make it setuid.

## General Installation

1) First, at a command line prompt, unpack the distribution (untar your .tar file). Don't do this in the directory where you want to install RT.

```
tar xzvf rt.tar.gz -C /tmp
```

or

```
cd /tmp; gzip -dc rt.tar.gz | tar xf -
```

You might also want to unpack into a place like /usr/src, so you have a copy of the install files (and the ability to uninstall) somewhere more permanent than in /tmp, which on some Unixes is wiped by default each boot.

2) From the temporary directory you just created, run the "configure" script.

To see a list of configuration options, use the "help" flag, which will show you a list of options:

```
./configure --help
```

We've also listed the options here. Defaults are specified in brackets.

### Configuration:

## DRAFT, April 13, 2003

-h, --help        display this help and exit  
--help=short     display options specific to this package  
--help=recursive display the short help of all the included packages  
-V, --version     display version information and exit  
-q, --quiet, --silent do not print `checking...` messages  
--cache-file=FILE cache test results in FILE [disabled]  
-C, --config-cache alias for `--cache-file=config.cache`  
-n, --no-create   do not create output files  
--srcdir=DIR      find the sources in DIR [configure dir or `..`]

### Installation directory:

By default, `make install` will install all the files in `/opt/rt3/bin`, `/opt/rt3/lib` etc. You can specify an installation prefix other than `/opt/rt3` using `--prefix`, for instance `--prefix=\$HOME`.

--prefix=PREFIX    install architecture-independent files in PREFIX  
                  [/opt/rt3]  
--exec-prefix=EPREFIX install architecture-dependent files in EPREFIX  
                  [PREFIX]

### Installation directory fine-tuning:

Fine tuning of the installation directories:

--bindir=DIR        user executables [EPREFIX/bin]  
--sbindir=DIR       system admin executables [EPREFIX/sbin]  
--libexecdir=DIR    program executables [EPREFIX/libexec]  
--datadir=DIR       read-only architecture-independent data [PREFIX/share]  
--sysconfdir=DIR    read-only single-machine data [PREFIX/etc]  
--sharedstatedir=DIR modifiable architecture-independent data [PREFIX/com]  
--localstatedir=DIR modifiable single-machine data [PREFIX/var]  
--libdir=DIR        object code libraries [EPREFIX/lib]  
--includedir=DIR    C header files [PREFIX/include]  
--oldincludedir=DIR C header files for non-gcc [/usr/include]  
--infodir=DIR       info documentation [PREFIX/info]  
--mandir=DIR        man documentation [PREFIX/man]

### Optional Features:

--disable-FEATURE   do not include FEATURE (same as --enable-FEATURE=no)  
--enable-FEATURE[=ARG] include FEATURE [ARG=yes]  
--enable-layout=LAYOUT Use a specific directory layout (Default: RT3)

### Optional Packages:

--with-PACKAGE[=ARG] use PACKAGE [ARG=yes]  
--without-PACKAGE    do not use PACKAGE (same as --with-PACKAGE=no)

## DRAFT, April 13, 2003

```
--with-rt-group=GROUP  group to own all files (default: rt)
--with-bin-owner=OWNER user that will own rt executable files (default root)
--with-libs-owner=OWNER user that will own RT libraries (default root)
--with-libs-group=GROUP group that will own rt binaries (default bin)
--with-db-type=TYPE    sort of database RT will use (default: mysql)
--with-db-host=HOSTNAME FQDN of database server (default: localhost)
--with-db-port=PORT    port on which the database listens on
--with-db-rt-host=HOSTNAME
                        FQDN of database server (default: localhost)
--with-db-dba=DBA      name of database administrator (default: root)
--with-db-database=DBNAME
                        name of the database to use (default: rt3)
--with-db-rt-user=DBUSER
                        name of database user (default: rt_user)
--with-db-rt-pass=PASSWORD
                        password for database user (default: rt_pass)
--with-web-user=USER   user the web server runs as (default: www)
--with-web-group=GROUP group the web server runs as (default: www)
```

### Some influential environment variables:

PERL Perl interpreter command

Use these variables to override the choices made by 'configure' or to help it to find libraries and programs with unusual names/locations.

Look at all the available flags and determine your RT install preferences, then:

```
./configure (with the flags you want)
```

The usage is:

```
./configure [OPTION]... [VAR=VALUE]...
```

3) Satisfy RT's myriad dependencies (in other words, make sure your system has all the stuff it needs to run).

3.1) First, check for compliance:

```
perl sbin/rt-test-dependencies \
    --with-<databasename> --with-<web-environment>
```

Instead of <databasename>, type either `mysql` or `postgres`. Instead of <web-environment>, type either `fastcgi`, `modperl1`, or `modperl2`

If there are unsatisfied dependencies (if you are missing stuff), you will see complaints like "such-and-such not installed." Go to step 3.2 below.

3.2) As root:

```
perl sbin/rt-test-dependencies \
    --with-<databasename> --with-<web-environment> --install
```

---

Note: if you've never run CPAN on your machine, run it first or this part of the install will not be fun.

```
perl -MCPAN -eshell
```

You'll be walked through the configuration process.

When you're done:

```
quit
```

---

---

Note: if things are still messed up, you are logged in as root, right?

---

Alternately, you could install missing items by hand. For example:

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

would install DBD::mysql

3.3) Check to make sure everything was installed properly:

```
perl sbin/rt-test-dependencies \  
    --with-<databasename> --with-<web-environment>
```

4) Create a new Unix group called 'rt'

5) For a **new installation** (If you're upgrading within the RT 3.0.x series, skip to step 7):

5.1) As root and from within the rt directory, type:

```
make install  
make initialize-database
```

5.2) If the make fails, type:

```
make dropdb
```

and start over from step 5.1

5.3) If make install still fails, look in /etc/httpd/httpd.conf or /etc/httpd/conf/commonhttpd.conf (or wherever your httpd.conf is; this may vary by Unix distribution) for a line that reads "Group [something]" and another that reads "User [somebody]"

Go to your RT source directory and type (without the brackets, substituting the right Group and User from the file you just looked at):

```
./configure --with-web-group=[something] --with-web-user=[somebody]
```

Skip to step 7.

6) For **upgrading** within the RT 3.0.x series -- if you are not upgrading or have already completed step 5, skip to step 7

As root, type:

```
make upgrade
```

(replace "make" with the local name for Make, if you need to)

This will build new executable files, config files and libraries without overwriting your RT database.

It may then instruct you to update your RT system database objects

7) Edit `etc/RT_SiteConfig.pm` in your RT installation directory, by specifying any values you need to change from the defaults. If `RT_SiteConfig.pm` is blank, start by copying the all the settings in `RT_Config.pm`.

In many cases sensible defaults have been included. In others, you must supply a value. Some values (such as the RT log directory) will come from values you supplied in the Makefile. You'll find further explanation inline in the `RT_SiteConfig.pm` file. You should look at and consider changing the following:

**`$DatabasePassword`** = 'rt\_pass'

password the DatabaseUser should use to access the database. Some MySQL users have had trouble with passwords of longer than 8 characters; if you cannot connect, try a password of 8 characters or fewer.

**`$CanonicalizeEmailAddressMatch`** = 'subdomain.example.com\$';

**`$CanonicalizeEmailAddressReplace`** = 'example.com';

The `$CanonicalizeEmailAddress` variables allow you to keep incoming messages consistent, such as when a site removes the subdomain from an email address. In the example presented by the defaults, if messages from your organization sometimes come from *subdomain..example.com* and sometimes from *example.com*, you'd set `$CanonicalizeEmailAddressMatch` to *exchange.example.com* and `$CanonicalizeEmailAddressReplace` to *example.com*

**`$SenderMustExistInExternalDatabase`** = undef;

If `$SenderMustExistInExternalDatabase` is true, RT will refuse to create non-staff accounts for unknown users if you are using the "LookupSenderInExternalDatabase" option elsewhere in `RT_SiteConfig.pm`. Instead, an error message will be mailed and RT will forward the message to `$RTOwner` as defined in `RT_SiteConfig.pm`. If you are not using `$LookupSenderInExternalDatabase`, this option has no effect. If you define an `AutoRejectRequest` template, RT will use this template for the rejection message.

**`$CorrespondAddress`** = 'RT::CorrespondAddress.not.set';

**`$CommentAddress`** = 'RT::CommentAddress.not.set';

`$CorrespondAddress` and `$CommentAddress` are the default addresses that will be listed in From: and Reply-To: headers of reply and comment mail tracked by RT, unless overridden by a queue-specific address.

**`$MailCommand`** = 'sendmailpipe';

`$MailCommand` defines which method RT will use to try to send mail. We know *sendmailpipe* works fairly well. If *sendmailpipe* doesn't work well for you, try *sendmail*. Note that you should remove the '-t' from `$SendmailArguments` if you use *sendmail* rather than *sendmailpipe*. Also note that *sendmailpipe* and *sendmail* aren't the names of commands on your system, but instructions that tell RT what mail delivery subsystem to try.

**`$SendmailArguments`** = "-oi -t";

`$SendmailArguments` defines what flags to pass to `$Sendmail`, assuming you picked *sendmail* or *sendmailpipe* as the `$MailCommand`. If you picked *sendmailpipe* you *must* add a -t flag to `$SendmailArguments`. These options are good for most sendmail wrappers and workalikes,.

**`$SendmailPath`** = "/usr/sbin/sendmail";

If you selected *sendmailpipe* as `$MailCommand`, you *must* specify the path to your sendmail executable file in `$SendmailPath`. If you did not select *sendmailpipe* this has no effect.

**`$Timezone`** = 'US/Eastern';

`$Timezone` is used to convert times entered by users into GMT and back again. It should be set to a timezone recognized by your local Unix box

**`$UseFriendlyToLine`** = 0;

RT can set a "friendly," rather than blank, `T0:` header when sending messages to Ccs or AdminCcs. This feature does not work with Sendmail(tm) brand sendmail. If you are using sendmail, rather than postfix, qmail, exim, or some other program, you must disable this option.

**\$WebPath** = "";

A variable used to help RT construct URLs that point back to RT. If you've put RT somewhere other than at the root of your webserver, you need to define a WebPath. RT uses this in the construction of relative URLs.

\$WebPath requires a leading / but no trailing /

**\$WebBaseURL** = "http://RT::WebBaseURL.not.configured:80";

A variable used to help RT construct URLs that point back to RT. \$WebBaseURL is the base of the URL. it should usually include the scheme, the host, and the port if non-standard. Such as "https://rt.example.com" or "http://rt.example.com:88"

\$WebBaseURL doesn't need a trailing /

**\$WebURL** = \$WebBaseURL . \$WebPath . "/";

A variable used to help RT construct URLs that point back to RT. \$WebURL is the combination of \$WebBaseURL and \$WebPath. Generally, you shouldn't change it.

**\$WebImagesURL** = \$WebURL."/NoAuth/images/";

\$WebImagesURL points to the base URL where RT can find its images. If you're running the FastCGI version of the RT web interface, you should make RT's `WebRT/html/NoAuth/images` directory available on a static web server and supply that URL as \$WebImagesUrl -- alternately, you can tell Apache not to run it through FastCGI.

## Setting Up The Web Interface

RT's web interface is based around `HTML::Mason`, which works best with the `mod_perl` perl interpreter within Apache httpd. Alternatively, support for the FastCGI (and plain CGI) interface is also provided as `'bin/mason_handler.fcgi'`.

## Configuring Apache

1) First, you'll need to add a few lines to your `httpd.conf` telling it to use RT's web interface. If you have `mod_perl` (you should, the Perl scripts will go quite a bit faster around with it), you can do something like this:

```
<VirtualHost your.ip.address>
  ServerName your.rt.server.hostname
  DocumentRoot /opt/rt3/share/html
  AddDefaultCharset UTF-8

  # this line applies to Apache2+mod_perl2 only
  PerlModule Apache2 Apache::compat

  PerlModule Apache::DBI
  PerlRequire /opt/rt3/bin/webmux.pl

<Location />
  SetHandler perl-script
  PerlHandler RT::Mason
```

```
</Location>  
</VirtualHost>
```

---

Note: If you are running Apache 2, the recommended configuration directive is:

```
RedirectMatch permanent (.*)$ http://myaddress.com$1/index.html
```

in your VirtualHost section

---

2) Next, stop Apache. This will vary by system but might look something like this:

```
cd /etc/init.d  
.httpd stop
```

3) Try to use a browser to go to your RT server's website. You shouldn't be able to do it.

4) Restart Apache

```
.httpd start
```

Once this works, you may move on to setting up your email gateway.

## Setting up the mail gateway

The mail gateway requires the web ui to be set up and functional. If you intend to use SSL to secure your mail gateway, you need to make sure that lib-www-perl, a.k.a LWP, is built with SSL support.

1) Documentation for setting up the mail gateway can be found by typing:

```
perldoc /<path to installed program>/rt-mailgate
```

The rt-mailgate program is installed in /opt/rt3/bin by default, so that command will probably be, more specifically:

```
perldoc /opt/rt3/bin/rt-mailgate
```

2) You need to tell your Mail Transfer Agent (ex sendmail, postfix, or qmail) how to forward messages to RT's mail gateway. To do this, create an aliases in your system's mail aliases file.

Add the following lines to /etc/aliases (or your local equivalent such as /etc/mail/aliases) :

rt:

```
"|/opt/rt3/bin/rt-mailgate --queue general --action correspond --url  
http://localhost/"
```

rt-comment:

```
"|/opt/rt3/bin/rt-mailgate --queue general --action comment --url  
http://localhost/"
```

---

Notes:

When you add the above it should be all one line; make sure you include the quotes.

You'll probably want to replace "general" with the name of a particular queue.

---

You'll choose from "action correspond" and "action comment" depending on whether the mail should be resent to the requestor. If you want the requestor to see it, choose *correspond*; if you want to keep it from the requestor, choose *comment*.

the URL will be the URL for RT's web interface, such as `http://rt3.fsck.com`, `http://foobar.example.com`, whatever is the root of your RT install.

---

- 3) Stop and start your webserver, so it picks up your configuration changes.
- 4) Change the password. Root's password for the web interface is *password*. Not changing this is a security risk.

For more information on configuring your mail gateway, see *Chapter 4: Adminning RT*.

## Configure RT

Until you configure RT, RT will not be able to send or receive email, nor will it be more than marginally functional. This is not an optional step. RT has many configuration options, and you should read *Chapter 4: Adminning RT* for a broader view, but here's a quick guide to things you should probably look at.

### Create users

In its default configuration, RT uses an internal users database to keep track of who can access RT and who has which 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.

click Configuration -> click Users -> click New user

For more details, see *Users: Creating a user* in *Chapter 4: Adminning RT*.

### Grant user rights

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 by email, grant the system group Everyone the rights See Queue, Create Ticket, Reply To Ticket, and Comment On Ticket for that queue.

If you intend to let ticket requestors use the requestor-mode web interface, you should grant the role of Requestor the right Show Ticket.

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

- Show Ticket
- Show Ticket Comments
- Watch
- Watch As AdminCc
- Own Ticket
- Modify Ticket

For more details, see *Users: Granting user rights* in *Chapter 4: Adminning RT*.

## Create queues

By default, RT has one queue called 'general'. You might want to change the values for this.

click Configuration -> click Queues -> click general

To create a new queue:

click Configuration -> click Queues -> click New queue

For more details, see Queues in Chapter 4: Adminning RT.

## Set up scrips

The default global Scrips installed with RT 3 are:

- On Create, Autoreply to Requestors with Global Template: Autoreply
- On Create, Notify AdminCcs with Global Template: Transaction
- On Correspond, Notify AdminCcs with Global Template: Admin Correspondence
- On Correspond, Notify Requestors and Ccs with Global Template: Correspondence
- On Correspond, Notify Other Recipients with Global Template: Correspondence
- On Comment, Notify AdminCcs as Comment with Global Template: Admin Comment
- On Comment, Notify Other Recipients as Comment with Global Template: Correspondence
- On Resolve, Notify Requestors with Global Template: Resolved

On Correspond, Open Ticket with Global Template: Blank (this scrip replaces old built-in behavior that would automatically open tickets if privileged users replied to a new ticket or if anyone replied to a stalled or closed ticket.)

For more details, see *Scrips* in *Chapter 4: Adminning RT* and *Appendix 3: Scrips*.

**DRAFT, April 13, 2003**

# Chapter 3

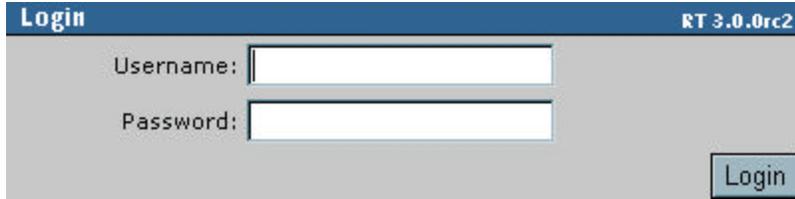
## Using RT on the Web

RT has a web interface that works in just about every web browser -- including screen readers for the blind -- and is easy to use. How easy? You're about to find out.

If you see any unfamiliar terms in this chapter, they should be defined in the *RT Concepts* section of Chapter 1: *Introduction* and/or *Appendix 1: Glossary*.

### Logging in

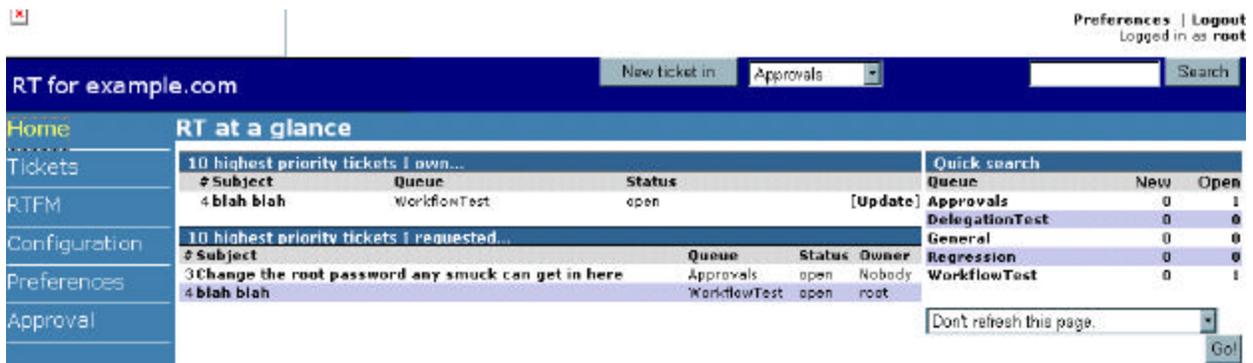
Your administrator will tell you where your login page is. The login interface looks like this:



type your username in the Username text box -> type your password in the Password text box -> click the Login button.

### Home page interface

Immediately after logging in, you'll be looking at the home page.



RT for example.com

New ticket in: **Approvals** Search

Preferences | Logout  
Logged in as root

#### Home

#### RT at a glance

10 highest priority tickets I own...			
#	Subject	Queue	Status
4	blah blah	WorkflowTest	open

10 highest priority tickets I requested...				
#	Subject	Queue	Status	Owner
3	Change the root password any smuck can get in here	Approvals	open	Nobody
4	blah blah	WorkflowTest	open	root

Quick search		
Queue	New	Open
Approvals	0	1
DelegationTest	0	0
General	0	0
Regression	0	0
WorkflowTest	0	1

Don't refresh this page. Go!

---

Note: You'll find that many items are separately clickable in the RT interface, such as ticket numbers and queues.

---

## RT at a glance

Let's start with the center section of the page on the white background.

**10 highest priority tickets I own:** This section displays what you need to do. Each ticket will display its number, subject, queue, and status from this page. Click any ticket to display its basic information page (see also the *Tickets interface* section of this chapter) or click `Update` to respond or comment on a ticket.

**10 highest priority tickets I requested:** Here, you'll see to see the progress made on tickets you've requested. Each ticket will display its number, subject, queue, status, and owner. Click any ticket for more detail.

**Quick search:** There'll be a list of queues under this heading. You'll be able to see how many tickets in each queue are new, open, and stalled. This is another way to see what needs to get done. Click on any queue for more detail.

**Don't refresh this page:** selected by default from a drop-down menu in the lower right. Other options are refreshing the page every 2, 5, 10, 20, 60, and 120 minutes. Click the `Go!` button to finalize the change.

## Top navigation bar

**New ticket in:** Use this to quickly start creating a ticket (for more detail, see the *Creating a new ticket* section in this chapter). Choose a queue from the drop-down menu and click the `New ticket in` button to begin.

**Search:** type a search term or a ticket number in the text box and click the `Search` button to find tickets containing that term.

**Preferences:** click this link to edit your personal information on RT. On a basic level, this is information such as email address, name, password, phone numbers, location, and signature. This is also where you start working with personal groups and delegating rights, which we'll explain later in this chapter.

**Logout:** click this to log out of RT. You can leave yourself logged into RT without breaking anything, but logging out is a good habit to get into for security purposes.

## Left side navigation bar

These are the basic options. They will expand as you perform various tasks. Your current location will always be colored yellow.

**Home:** where you are now. You can always get back to the home page by clicking `Home` in the upper left of the screen.

**Tickets:** clicking this brings you to the ticket search page (see also the *Common tasks: Searching* section of this chapter).

**RTFM:** clicking this takes you to RTFM, RT's integrated knowledgebase (documented separately).

**Approval:** sometimes, a ticket may require the approval of someone in your organization before it can be resolved. This section shows the status (pending, approved, denied) of approval requests. Most users won't need to worry much about Approvals; there is more detail in *Chapter 4: Adminning RT*.

**Configuration:** for administrative staff performing higher-level tasks like managing users, groups, and queues; see *Chapter 4: Adminning RT* if this applies to you. Most users won't want to (and will not have the right to) change anything here.

**Preferences:** click this link to edit your personal information on RT. On a basic level, this is information such as email address, name, password, phone numbers, location, and signature; this is also where you start working with personal groups and delegating rights. (See *Chapter 4: Adminning RT* if the more advanced tasks apply to you).

# Tickets interface

First, you should know some of the various ways to get "inside" an existing ticket:

- from the Home page, click a ticket name under `10 highest priority tickets I own` or `10`

highest priority tickets I requested

- search for a ticket, from the Home page or by clicking **Tickets** in the left side navigation bar. Then click the ticket.
- click a queue and then click a ticket.
- clicking a ticket's URL from within an email message

## Display

Any way you get inside a ticket, you'll arrive at that ticket's **Display** page first.

The screenshot shows the RT interface for ticket #5. At the top, there's a navigation bar with 'Home' and '#5: screen shot!'. Below that, a blue sidebar contains navigation links: Tickets, Search, New Search, #5 (highlighted), Display, History, Basics, Dates, People, Links, Jumbo, RTFM, Configuration, Preferences, and Approval. The main content area is divided into several sections: 'The Basics' (Id: 5, Status: new, Worked: 0 min, Priority: 0/0, Queue: General), 'Custom Fields' (TestingCF: (no value)), 'People' (Owner: Nobody, Requestors: root@localhost, Cc:, AdminCc:), 'Dates' (Created: Fri Apr 11 10:57:34 2003, Starts: Not set, Started: Not set, Last Contact: Not set, Due: Not set, Closed: Not set, Updated: Fri Apr 11 10:57:50 2003 by root), and 'Relationships' (Depends on:, Depended on by:, Parents:, Children:, Refers to: 4: (root) blah blah [open], Referred to by:). At the top right of the main area, there are links: Reply | Resolve | Open | Take | Comment | Extract Article.

The Display page (pictured above) provides an overview of the ticket. As you can see, it includes information like Basics, Dates, Custom Fields, Relationships/Links, and People (we'll go into more detail about that information later). The ticket's History, while not visible in this screenshot, is also available on the Display page (see next subsection, *History*). Note the ticket's number and name near the top of the screen. The ticket's number is also in yellow in the left side navigation bar.

Ticket pages will generally offer the options **Reply**, **Resolve**, **Open**, **Take**, **Comment**, and **Extract Article** in the upper right. **Reply** and **Comment** are covered in the *Replying and Commenting* section of this chapter, **Resolve** means closing a ticket because work on it is finished, **Open** will change a ticket's status to "open," **Take** makes you the ticket's owner, and **Extract Article** is part of RT's integration with the knowledgebase RTFM. Once a ticket is taken, the **Take** option disappears; likewise, once a ticket is resolved, the **Resolve** option disappears. When a resolved ticket is reopened with the **Open** option, **Open** disappears and **Resolve** reappears.

Often, a block of information on the Display page will have its own ticket page, which you can reach from the Display page by using the clickable section headers within the ticket display like Basics, Dates, etc. You can also always reach a ticket page from within a ticket by clicking the correct item (Basics, Dates, etc.) in the left side navigation bar. We'll start explaining those options soon.

The Display page sometimes has an **Attachments** box (not a hyperlink), which lists ticket attachments, organized by name and then by date.

## History

This page shows everything that's ever happened to the ticket, including comments, replies, and addition of attachments.

RT for fsck.com      New ticket in: cpan-questions      Search

Home      **Ticket History # 2089 no way to edit custom fields**      Reply | Comment

**Tickets**  
 Search  
 << First  
 < Prev  
 New Search

**#2089**  
 Display  
**History**  
 Basics  
 Dates  
 People  
 Links  
 Jumbo

Configuration  
 Preferences  
 Approval

**History**      Display mode: [Brief headers] [Full headers]

Wed Feb 12 19:40:24 2003      [Redacted] - Ticket created      [Reply] [Comment]  
 To: rt-3.0-bugs@fsck.com  
 Subject: no way to edit custom fields  
 From: [Redacted]  
 Date: Wed, 12 Feb 2003 16:40:22 -0800  
 Download (untitled) 2348

(rt 2.1.70)

if I create a multiple value custom field for a queue, there's no way to edit it's entries. I have to delete and recreate values to fix a typo. I'm sure this would be bad for things already assigned to a keyword...

Thu Feb 13 17:21:25 2003      jesse - Correspondence added      [Reply] [Comment]  
 Download (untitled) 5348

> [Redacted] - Wed Feb 12 19:40:24 2003:  
 > (rt 2.1.70)  
 >  
 > if I create a multiple value custom field for a queue, there's no way  
 > to edit it's entries. I have to delete and recreate values to fix a  
 > typo. I'm sure this would be bad for things already assigned to a  
 > keyword...

Actually, it's designed with loose referential integrity. deleting an available value won't touch existing tickets. Values are associated with tickets by name, not reference.

> [Redacted]  
 >  
 >  
 >  
 Jesse R. Vincent - jesse@bestpractical.com  
 Thu Feb 13 17:21:26 2003      jesse - Status changed from new to resolved

## Basics

This is where you change the ticket's subject, status, time worked, time left, priority, final priority, and queue. A ticket's subject is what it sounds like; "time worked" and "time left" refer to the number of minutes worked on this ticket only. If you're unclear on status, priority, or queue, refer to the *RT Concepts* section of *Chapter 1: Introduction*.

After changing information on the Basics page, don't forget to click the `Save Changes` button.

This is also the page where you can change values in a **custom field**. Your organization may create its own fields, and they come up on a ticket's `Basics` page. They work like any other field: you type things in them or select things from them. To create your own custom fields, see *Chapter 4: Adminning RT*.

## Dates

From this page, you can adjust several dates relating to a ticket.

**Starts:** the date work should begin on a ticket (optional)

**Started:** the date work began on a ticket

**Last Contact:** the date the ticket was last modified

**Due:** the date by which you must resolve the ticket and have a pint

To change any of these dates, type the new date in the text box and click the `Submit` button.

---

Note: Acceptable date formats such as mm/dd/yy or dd/mm/yyyy will vary by organization and country. If you're not sure, ask an administrator how to enter dates. Administrators can consult <http://search.cpan.org/author/MUIR/Time-modules-2003.0211/lib/Time/ParseDate.pm> for a date parsing perl module

---

## People

This page deals with a ticket's owner, requestor, Cc(s), and Admin Cc(s). To review these roles, see the *RT Concepts* section of *Chapter 1: Introduction*.

To reach a ticket's People screen

from within a ticket, click People in the left side nav bar

The screenshot shows a web interface for managing ticket watchers. The page title is "Modify people related to ticket #5". A left-hand navigation menu includes "Home", "Tickets", "Search", "New Search", "#5", "Display", "History", "Basics", "Dates", "People" (highlighted), "Links", "Jumbo", "RTFM", "Configuration", "Preferences", and "Approval". The main content area is titled "Modify people related to ticket #5" and contains several sections: "New watchers" with search criteria for "User Id" and "Name", "Owner" dropdown set to "Nobody", and "Current watchers" with a list of requestors including "root" and "none". There are also "Go!" buttons for the search fields.

There are two ways to **add a watcher**. If you know a potential watcher's email address, you can add one directly. Otherwise, you can search for them, then add whoever you've found.

To **search for users** before adding them as watchers:

look under Find people whose -> choose User Id, Email, Name, or Organization from the first drop-down menu -> choose contains, doesn't contain, is, or isn't from the second drop-down menu -> type your search term in the text box -> click the Go! button

If RT finds a user, her username will appear in the Add new watchers area, the lower half of the gray box. You've now found your user, and you're ready to configure her as a watcher.

To **search for groups** before adding them as watchers:

look under Find group whose -> choose Name or Description from the first drop-down menu -> choose contains, doesn't contain, is, or isn't from the second drop-down menu -> type your search term in the text box -> click the Go! button

If RT finds a group, the group name will appear in the Add new watchers area, the lower half of the gray box. You've now found your group, and you're ready to configure it as a watcher.

To **add new watchers**:

locate the Add new watchers section of the page -> for each watcher you're adding, choose Requestor, Cc, or AdminCc from the Type drop-down menu -> type the new watcher's email address in the Email text box, if it's not already filled in -> click the Save Changes button

For each email address, RT will search for an existing user to match it. If there is no match, RT will create a new non-staff user. (If the user is already filled in from a search, it won't be automatically created)

**To remove a watcher:**

locate the `Current Watchers` section of the page -> check the box next to the watcher(s) to delete -> click the `Save Changes` button

**To assign a ticket or change its owner:**

change `Owner` drop-down menu to the person of your choice -> click the `Save Changes` button

To **change a ticket's owner to yourself**, click `Take` from within a ticket. Generally, you can only take a ticket that is owned by `Nobody`. There is no intermediate "Are you sure you want to do this?" page, but if you wrongly take a ticket, you can always give it back. Once a ticket is taken, the `Take` option disappears from the ticket's interface.

**Relationships/Links**

Relationships in RT include `Depends on`, `Depended on by`, `Parents`, `Children`, `Refers to`, and `Referred to by`. If we just lost you, see the *RT Concepts* section of *Chapter 1: Introduction*. For the sake of space, relationships are sometimes called **Links** in RT, such as in the left side navigation bar from within a ticket.

To reach a ticket's Relationships page:

from within a ticket, click `Links` in the left side nav bar

You can add or delete relationships here, or merge duplicate tickets. In this screen shot, the relationships/links page for ticket #4, you'll see that ticket #4 refers to ticket #1.

**Link ticket #4** Reply | Resolve | Open | Take | Comment | Extract Article

**Edit Relationships**

<b>Current Relationships</b> <small>(Check box to delete)</small> Depends on: Depended on by: Parents: Children: Refers to: <input type="checkbox"/> <b>1: (root) Sample workflow test [new]</b> Referred to by:	<b>New Relationships</b> <small>Enter tickets or URIs to link tickets to. Separate multiple entries with spaces.</small> Merge into: <input type="text"/> <small>(only one ticket)</small> Depends on: <input type="text"/> Depended on by: <input type="text"/> Parents: <input type="text"/> Children: <input type="text"/> Refers to: <input type="text"/> Referred to by: <input type="text"/>
---	--

**Save Changes** Submit

**To merge two identical, duplicate tickets:**

look under `New Relationships` -> enter only one ticket number in the `Merge Into:` text box -> click the `Submit` button

If someone tries to use the ticket ID of a ticket you've merged into another one, RT automatically redirects to the correct, newly-merged ticket.

**To add a relationship**(`Depends on`, `Depended on by`, `Parents`, `Children`, `Refers to`, `Referred to by`):

look under `New Relationships` -> enter ticket numbers (or other data) into the appropriate fields, separating multiple entries with spaces -> click the `Submit` button

**To delete a relationship**

look under `Current Relationships` -> check the box next to the relationship to delete -> click the `Submit` button

To add a **relationship to an RTFM article**, see *Replying and Commenting* in this chapter.

## Jumbo

This page shows editable versions of the Basics, Dates, People, and Relationships screens on one page, plus Update Ticket fields for adding comments or replies. This screen is useful if you must modify several aspects of a ticket at once. Be sure to click the Save Changes button when you're done.

---

Note: For the rest of this chapter, when we say something like "from within the ticket's History page..." that means you should click History to get there, if you're not there already. If you don't see a History link to click, that likely means you're not in the ticket yet. If you're already in the History page, you don't have to click History again, of course. See the *Tickets interface* section earlier in this chapter if this is confusing you.

---

## Common tasks

Now that you know your way around RT's web interface, there are probably many tasks you can figure out on your own. However, we wanted to make sure you had a step-by-step guide to the most important and common tasks such as using RT's search functionality and working with tickets.

## Searching

Search in RT is fairly intuitive, but there are some features worth noting:

- Searches, by default, are sticky. That is, if you return to the Search page during a session, it will display the most recent search.
- Searches are, by default, limited to 50 results per page.
- Searches are iterative -- you build them up. If you search for tickets owned by Jane, then search for open tickets, you'll get open tickets owned by Jane.
- Iterative searches do work with incompatible conditions, that is, x OR y. If you search for tickets owned by Jane, and then for tickets owned by Alice, you'll get tickets owned by Jane OR Alice. Thanks to Lorens Kockum for pointing this out.
- When RT displays multiple search results, it will also display Previous, Next, First, and Last links for navigating through the matching tickets.
- Clicking queue links takes you to a search page that displays search results for new and open tickets in that queue. That search can then be further refined.

### Simple ticket search:

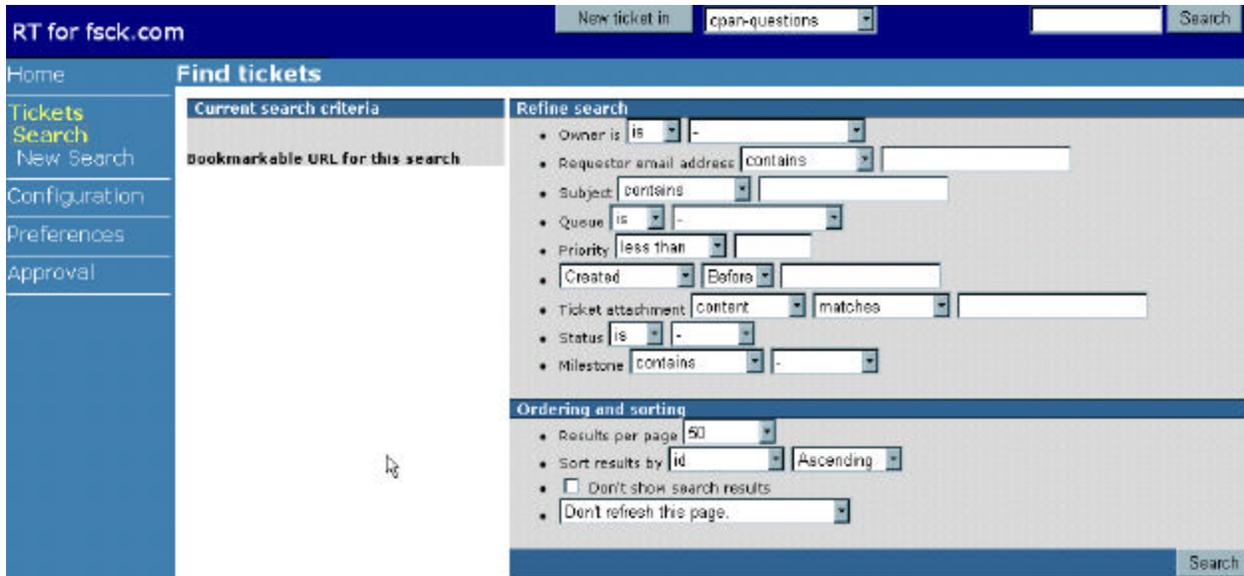
type your search term in the text box in the upper right of the screen -> click the Search button

If you get too many results, you can still pinpoint what you need by searching again to narrow the results down.

### More advanced ticket search:

click Tickets in the left side navigation bar

You'll be presented with a screen like this (the screen for a simple ticket search will look like this, but with search results at the top):



You can search by owner, requestor email address, ticket subject, queue, priority, attachment, and/or status. For instance, you could **search for tickets you own** by selecting your own name from the Owner drop-down menu, or search for only high priority tickets.

You may also alter the number of search results returned per page, how the results are sorted, and how often the page is refreshed. Click the Search button when you're done.

To **bookmark a search** so you can return to the results later:

click Bookmarkable URL for this search -> add the page to your Bookmarks, Favorites, or whatever applies for your browser

To clear previous search results and start a **new search**:

click New Search in the left side navigation bar

To **search for new and open tickets in a queue**:

from the Home page, click a queue name under Quick search

You'll be taken to a search page populated with a search for new and open tickets in that queue. Enter more search terms to refine the search.

To **search for users**:

from within a ticket's People page, look under Find people whose -> choose User Id, Email, Name, or Organization from the first drop-down menu -> choose contains, doesn't contain, is, or isn't from the second drop-down menu -> type your search term in the text box -> click the Go! button

To **search for groups**:

from within a ticket's People page, look under Find group whose -> choose Name, or Description from the first drop-down menu -> choose contains, doesn't contain, is, or isn't from the second drop-down menu -> type your search term in the text box -> click the Go! button

## Creating a new ticket

To **create a new ticket**, start by selecting its queue:

from any page, select the queue of your choice from the New ticket in drop-down menu near the top of the screen -> click the New Ticket in button

The next page will be the Create a new ticket page, defaulting to the Show basics view with the following fields:

**Queue:** this should already be filled in for you.

**Status:** generally, you'll leave this at the default, `new`. However, you might sometimes need to create a ticket to record a past occurrence or a problem that's already being worked on, so you might choose one of the other options from the drop-down menu.

**Owner:** you can give the ticket an owner or leave this blank and let someone claim it later.

**Requestors:** the default email address will be yours. Fill in any others if needed.

**Cc** and **Admin Cc:** if you don't know what these are for, see the *RT Concepts* section in *Chapter 1: Introduction*.

**Subject** and **Attach file** should look familiar to most email users and work basically the same way.

**Describe the issue below** means you should explain why this ticket exists.

If you click `Show details` or scroll to the bottom of the page, you'll be brought to the optional but potentially useful fields **Priority**, **Final Priority**, **Time Worked**, **Time Left**, fields for dates (**Starts** and **Due**), and fields for each relationship (**Depends on**, **Depended on by**, **Parents**, **Children**, **Refers to**, **Referred to by**). These, too, are explained in the *RT Concepts* section in *Chapter 1: Introduction*.

When you're done filling in fields, click the `Create` button to make the ticket.

Clicking `Show Basics` brings you back to the top of the page.

## Replying and commenting

There are two ways you can respond to a ticket. By default, a **reply** is public and a **comment** is hidden from the requestor. Comments and replies are listed permanently under the ticket's `History`.

To say something about a ticket, first make sure you are within one of the ticket's pages (`Display`, `History`, `Basics`...) then click `Comment` or `Reply` in the upper right of the page. To quote the text of an existing comment or reply, click `Reply` or `Comment` next to an existing reply or comment, from within the ticket's `History`, to quote the text of the existing reply or comment inline.

You'll see drop-down menus for `Status` and `Owner`. You probably won't need to change the defaults here.

There will also be a text box for how many minutes the ticket has been worked on. Fill this in if you want to or if you are supposed to be logging your time.

The `Update Type` drop-down menu will be automatically set to `Comments (Not set to requestors)` or `Response to requestors`, depending on whether you clicked `Comment` or `Reply` as your first step. You can select the other option if you've changed your mind.

As for the other fields, we'll assume you have a working knowledge of email. RT correspondence has some fields that should be familiar to email users, such as `Subject`, `CC`, `BCC`, `Attachments`, and `Message`. The `Subject` field may be automatically filled in with the name of the ticket.

---

Note: email addresses in the `CC` and `BCC` fields of replies and comments will not automatically receive future updates.

---

Enter your comment or reply in the `Message` field, and when everything's done, click the `Submit` button

You'll be brought back to the ticket's `Display` page, with a status message on top that says your comment or reply has been sent/recorded, or that there is a problem and it hasn't.

**To add a signature** that will appear by default at the bottom of all your comments and replies:

click `Preferences` in the upper right, or in the left side navigation bar -> type your (preferably short) signature in the `Signature` field -> click the `Submit` button

### Reply and comment integration with RTFM

RTFM is RT's integrated knowledgebase system. If you are using RTFM, you should know that when you reply to or comment on tickets in RT, you can search for and include (editable) RTFM articles in the reply or comment. The ticket will now have a new relationship: `Refers To [article of your choice]`.

## DRAFT, April 13, 2003

This screen shot shows the relevant part of the Reply or Comment screen, located under `Attach` in the `Message` portion of the page:

Message: Search for RTFM articles matching   
Include RTFM article:

If you know the name or number of the article you'd like to include, enter it in the `Include RTFM article` text box and click the `Go` button.

To find an RTFM article to include in the reply or comment, enter a search term in the `Search for RTFM articles matching` text box and click the `Go` button. Assuming there are search results, click the `Go` button next to the article you'd like to include.

## Modifying a ticket

Listed alphabetically by item that could be changed:

**due date** (or any other date):

from within a ticket's `Dates` page, type the new date in the `Due :` text box -> click the `Submit` button

**final priority:**

from within a ticket's `Basics` page, type a new number in the `Final Priority` field -> click the `Save Changes` button

**history:** we're sorry, a ticket's history cannot be changed.

**owner:**

from within a ticket's `People` page, change `Owner` in drop-down menu to the person of your choice -> click the `Save Changes` button

**priority:**

from within a ticket's `Basics` page, type a new number in the `Priority` field -> click the `Save Changes` button

**queue:**

from within a ticket's `Basics` page, select a new item from the `Queue` drop-down menu -> click the `Save Changes` button

Note that when you change a ticket's queue, queue-specific values such as custom fields will disappear. They will not be deleted, but they will be hidden.

**status:**

from within a ticket's `Basics` page, select a value from the `Status` drop-down menu -> click the `Save Changes` button

**subject:**

from within a ticket's `Basics` page, type new data in the `Subject` field -> click the `Save Changes` button

**time left:**

from within a ticket's `Basics` page, type a new number in the `Time Left` field -> click the `Save Changes` button

**time worked:**

from within a ticket's `Basics` page, type a new number in the `Time Worked` field -> click the `Save Changes` button

## Modifying people

### To assign a ticket or change its owner:

from a ticket's `People` page, change `Owner` in drop-down menu to the person of your choice -> click the `Save Changes` button

### To add new watchers:

from a ticket's `People` page, locate the `Add New Watchers` section of the page -> choose `Requestor`, `Cc`, or `AdminCc` from the `Type` drop-down menu -> type the new watcher's email address in the `Email` text box -> add up to two more watchers, for a total of up to three -> click the `Save Changes` button

### To remove a watcher:

from a ticket's `People` page, locate the `Current Watchers` section of the page -> check the box next to the watcher(s) to delete -> click the `Save Changes` button

## Modifying relationships

### To add a relationship (Depends on, Depended on by, Parents, Children, Refers to, Referred to by):

from within a ticket's `Relationships` (or `Links`) page, look under `New Relationships` -> enter ticket numbers (or other data) into the appropriate fields, separating multiple entries with spaces -> click the `Submit` button

### To delete a relationship

from within a ticket's `Relationships` (or `Links`) page, look under `Current Relationships` -> check the box next to the relationship to delete -> click the `Submit` button

### To merge a duplicate ticket into an identical ticket:

from within a ticket's `Relationships` (or `Links`) page, look under `New Relationships` -> enter only one ticket number in the `Merge Into:` text box -> click the `Submit` button

If someone tries to use the ticket ID of a ticket you've merged into another one, RT automatically redirects to the correct, newly-merged ticket. Merging tickets also merges all ticket data such as watchers, relationships, and history.

## Updating many tickets at once

Sometimes, you may need to make a change to several tickets at once. There's an easy way to do this:

locate the tickets you want (see the `Searching` section of this chapter for help) -> click `Update all these tickets at once`

You'll be sent to a `Bulk ticket update` page where you can add a comment or reply to all the tickets, or change the tickets' owner, requestors, `Cc`, `Admin Cc`, subject, priority, queue, status, start date, due date, and resolved date. The power!

Click the `Submit` button when you're done.

## Resolving a ticket

If a ticket has been completed and doesn't need any more work, you can resolve it by clicking `Resolve` in the upper right of one of the ticket's pages (`Display`, `History`, `Basics`...).

This takes you to a comment form (see above *Replying and Commenting* section) but sets the `Status` drop-down menu to `Resolved`.

You're encouraged to use the `Message` text box to explain why you're resolving the ticket.

When you're done, click the `Submit` button.

You'll be brought to the ticket's `Display` page and told the comment has been recorded and ticket has been resolved (or not, if there is a problem).

## Changing your own user information

To **change your own name, email address, phone numbers, location, or signature:**

click Preferences in the upper right of the screen or in the left side navigation bar -> type new information in the desired fields ->click the Submit button

To **change your own password:**

click Preferences in the upper right of the screen or in the left side navigation bar -> type the new password in the New Password text box -> type it again in the Retype Password text box -> click the Submit button

### Personal groups

Users can grant their rights to others by creating a personal group and delegating the rights to the members of that group. This is convenient if you're going on vacation or need to pass the buck for any reason.

To **make a new personal group:**

click Preferences ->click Personal Groups ->click New group in the left side navigation bar -> type your group's name and description in the Name: and Description: text boxes -> click the Submit button

To **add members to a personal group:**

click Preferences -> click Personal Groups -> click the name of a personal group -> click Members in the left side navigation bar -> select user(s) or group(s) from the Add members box -> click the Submit button

To **delete members from a personal group:**

click Preferences -> click Personal Groups -> click the name of a personal group -> click Members in the left side navigation bar -> check the box next to the user(s) or group(s) to delete -> click the Submit button

To **change a personal group's name or description:**

click Preferences -> click Personal Groups -> click the name of a personal group -> make changes -> click the Submit button

To **disable a personal group:**

click Preferences -> click Personal Groups -> click the name of a personal group -> clear the Enabled checkbox -> click the Submit button

Unlike disabled users and queues, disabled personal groups will still be visible in listings. That removes a step from the process of re-enabling them when necessary.

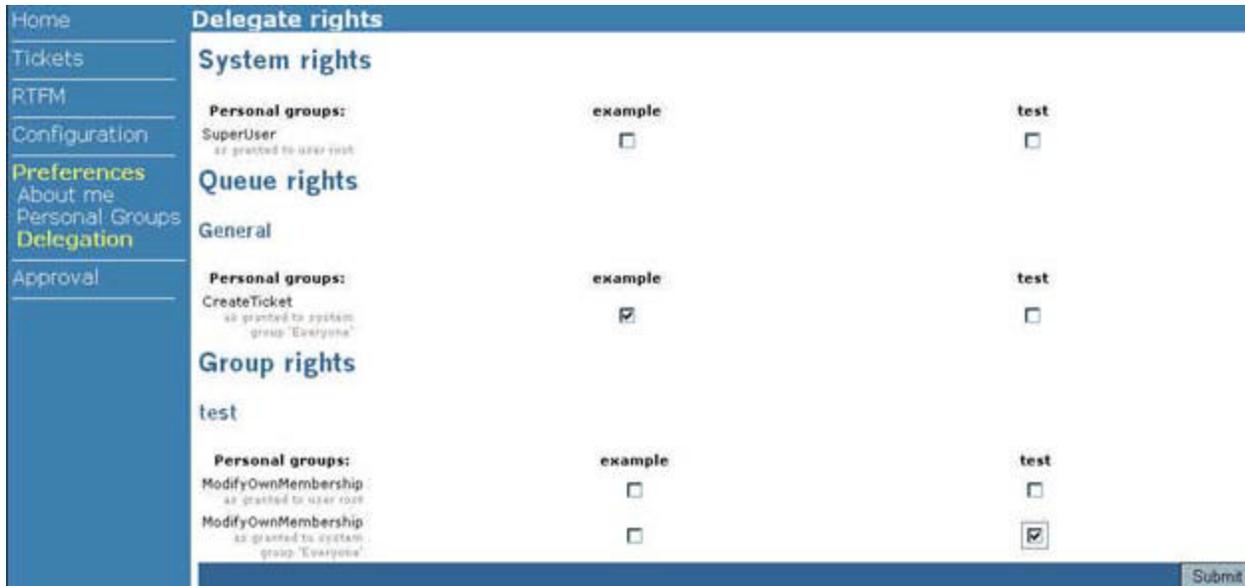
To **re-enable a personal group:**

click Preferences -> click Personal Groups -> click the name of a personal group -> checks the Enabled checkbox -> click the Submit button

### Delegating

Once you have a personal group, you can delegate rights.

- In order to delegate rights, you'll need to have the right "Delegate Rights."
- If you've been granted a right and you delegate it to someone, then someone revokes your right, that also revokes your delegate's right.
- If you've been granted a right globally, you can't delegate it for just one queue and vice versa.



**To delegate rights to a personal group:**

click Preferences -> click Delegation -> locate the right(s) you'd like to delegate -> check the box next to the right(s) and under the correct personal group(s)

**To stop delegating rights to a personal group:**

click Preferences -> click Delegation -> locate the correct right(s) -> clear the checkbox next to the right(s) and under the correct personal group(s)

## Logging out

You can leave yourself logged into RT without breaking anything, but logging out is a good habit to get into for security purposes. Logging out is standard:

click Logout in the upper right hand corner of the page.

Some RT configurations don't have a Logout button, so don't panic if you don't see one.

**DRAFT, April 13, 2003**

# Chapter 4

## Adminning RT

In this chapter, you'll learn how to use RT's built-in features to control users, groups, queues, and sometimes the entire system.

A word about the instructions in this chapter. Clicking the `Configuration` menu item will take you to the main Admin page, where you can then choose from `Users`, `Groups`, `Queues`, and `Global`. So most of our clickpaths in this chapter start with the step "click `Configuration`," but that step isn't always necessary. If you've already clicked `Configuration` and then `Users`, for example, you won't need to click them both again to admin users. Soon it will be obvious which steps you can skip and when, but our clickpaths all start from the beginning just to be as explicit as possible and provide instructions that can be used by anyone from anywhere in RT.

Before you start performing specific tasks, you should understand a few adminning concepts. Read the *Security*, *Rights*, *Custom fields*, *Scripts*, and *Approval* sections below, and the beginning of each of the sections *Users*, *Groups*, and *Queues*.

### Security

Basically, the usual rules of computer security apply:

- Don't put sensitive information (like credit card numbers) in any non-encrypted place like RT or your email program.
- Be aware that RT stores passwords in its database in an encrypted form.
- Also be aware that RT often sends unencrypted email containing replies and comments.
- RT lets you set a minimum password length for your users. Look for a value called `MinPasswordLength` in your configuration file.

Use the **SSL** (secure sockets layer) protocol for your servers; RT works fine over SSL. If you want to use **HTTPS** (HTTP over SSL) to secure traffic from the mail gateway to the web server, you'll need to install an additional package, the `libwww-perl` package, which RT's mail gateway uses to talk to the RT server.

We now quote from `readme.ssl` in the `libwww-perl` package:

The `libwww-perl` package has support for using `SSL/TLSv1` with its `HTTP` client and server classes. This support makes it possible to access `https` schemed URLs with `LWP`. Because of the problematic status of encryption software in general and certain encryption algorithms in particular, in several countries, `libwww-perl` package doesn't include `SSL` functionality out-of-the-box.

Encryption support is obtained through the use of `Crypt::SSLeay` or `IO::Socket::SSL`, which can both be found from CPAN. While `libwww-perl` has "plug-and-play" support for both of these modules (as of v5.45), the recommended module to use is `Crypt::SSLeay`.

## Rights

A right is permission for a user or group to do something in RT.

- Rights can be granted for individual groups, for individual queues, or across the whole system.
- Rights can be given to groups, roles (requestor, owner, Cc, AdminCc), system groups (everyone, staff users, non-staff users) or individual users.
- Rights can be delegated. See the *Delegating rights* section of this chapter.

It sounds complicated, but just ask yourself these questions:

- Who do you want to get the right? A group of users, one user? All ticket owners, all staff users, everybody?
- Where do you want the right to apply? A queue or all queues? Or in the case of a group right, one group or all groups?

We give step-by-step directions for granting (and revoking) rights throughout this chapter in the beginning of the sections *Users*, *Groups*, *Queues*, and *Global*. To delegate rights, see the *Delegating Rights* section in *Chapter 3: Using RT on the Web*.

## Custom fields

Custom fields are data fields that you can make up according to your organization's needs. You can create them for particular queues or globally. They appear on a ticket's `Display` page under the heading `Custom fields`.

There are four general types of custom fields:

**Select one value:** the user can select one value. The field is rendered as a box.

**Select multiple values:** the user can select one or several values. The field is rendered as a box.

**Enter one value:** the user can type one value. The field is rendered as a text box.

**Enter multiple values:** the user can type one or several values separated by line breaks. The field is rendered as a text box.

---

Note: Whenever you encounter a listing of custom fields (for a queue or globally), check the `Include disabled custom fields in listing` checkbox to view custom fields that have been disabled and are inactive.

---

## Scripts

Our business people make us call scrips "custom business logic." In English, this means that when Thing A happens, you can have RT do Task B for Important People C. In short, it's useful.

A scrip contains three parts:

**condition:** something that has to happen to activate the scrip, like a ticket being created.

**action:** the thing the scrip does in response to the condition, like notify the ticket's requestor.

**template:** essentially a form letter that goes along with an action. A simple example could be, "Dear [name of requestor], your ticket has been created and its ticket ID is [this number]."

Scripts can apply to individual queues or globally, but an individual ticket can't have its own scrip.

The default global Scripts installed with RT 3 are:

- On Create, Autoreply to Requestors with Global Template: Autoreply
- On Create, Notify AdminCcs with Global Template: Transaction
- On Correspond, Notify AdminCcs with Global Template: Admin Correspondence
- On Correspond, Notify Requestors and Ccs with Global Template: Correspondence
- On Correspond, Notify Other Recipients with Global Template: Correspondence
- On Comment, Notify AdminCcs as Comment with Global Template: Admin Comment
- On Comment, Notify Other Recipients as Comment with Global Template: Correspondence
- On Resolve, Notify Requestors with Global Template: Resolved

On Correspond, Open Ticket with Global Template: Blank (this scrip replaces old built-in behavior that would automatically open tickets if privileged users replied to a new ticket or if anyone replied to a stalled or closed ticket.)

See *Appendix 3: Scripts* for more detail.

## Approval

RT provides infrastructure for approvals and signoff for processes within your organization. RT Approvals are a useful way to get tickets OK'd by management without creating a new system. A supervisor's approval can be a ticket; another ticket can depend on that ticket being resolved. For instance, let's say you and your coworkers need permission from the CEO before you can make capital purchases. You can set up RT so that tickets in the Capital Purchases queue create an approval ticket that sends email to the CEO, who will then click `Approval` in the left side navigation bar to give the thumbs up. For an example of the custom programming required to create approvals, see *Appendix 5: Creating approvals*.

Most people won't ever need to click `Approval`, but we like to keep bosses happy, so here is the Approval screen:

The screenshot shows the 'My approvals' page in RT. On the left is a navigation menu with 'Approval' selected. The main area shows two approval items, each with 'Approve', 'Deny', and 'No action' (selected) options and a 'Notes' field. Below is a search section with checkboxes for 'Show pending requests' (checked), 'Show approved requests', 'Show denied requests', and 'Show requests awaiting other approvals'. There are also two text boxes for filtering by date: 'Only show approvals for requests created before' and 'Only show approvals for requests created after'.

It is basically a search page. Check any of the checkboxes, or enter dates in the text boxes, to narrow down the list of approval items you'd like to view. Usually, you will probably want to leave the default checkbox, `Show pending requests`, checked so you can see the list of tickets that should be approved or rejected. Then click the `Submit` button (not pictured) to see the results.

## Users

A user is a person who is in any way involved with RT. There are two general types of users:

**staff:** people who get to do things in RT.

**non-staff:** generally people outside your organization, like customers. Whenever someone new submits a request, RT automatically creates a non-staff account for that person. Non-staff users don't generally have any rights within RT, unless their role (such as "requestor") has been granted rights.

---

Note: You can't delete users, only disable them. So **disabled users** are also a user group of sorts. See also the *Disabling a user* subsection below.

---

### Creating a user

Sometimes it may be necessary to **add a user** to RT:

click `Configuration` -> click `Users` -> click `New user` in the left side navigation bar

At this time, you'll be on the `Create a new user` page, which offers many options for this new user. The only one that's mandatory is choosing a username, though filling in other fields would be useful. Here is a list of

headings and which fields you'll find under each of them:

**Identity:** text fields for Username, Email, Real Name, Nickname, Unix Login, and Extra Info.

**Location:** text fields for Organization, Address1, Address2, City, State, Zip, and Country.

**Access Control** has some options that require comment:

**Let this user access RT** checkbox: This is checked by default and means the user has not been disabled.

**Let this user be granted rights** checkbox: This is unchecked by default. Check the box to make this a staff user.

**New Password** and **Retype Password** text boxes: Each user needs a password, and RT won't randomly assign one. If you leave these fields blank, the user's account will be locked and she won't be able to do anything that requires a password -- that is, just about anything other than sending and receiving ticket-related email from her regular email account.

**Phone numbers:** text fields for Home, Work, Mobile, and Pager.

**Comments about this user:** a text field for any information not already entered in other fields.

Click the Submit button when you're done.

---

Note: You may find that when you add a user, you get an error "Name in use". This is probably because you have sent test emails when configuring the mail gateway. Click on "Users" and enter the search for the username or email address, making sure to check "Include disabled users in search."

---

### Searching for a user

To find an RT user:

click Configuration -> click Users -> in the search fields at the bottom of the page, select the proper options from the drop-down menu and type the search term you need -> check the Include disabled user in search box only if necessary (for instance, to restore an account) -> click the Go! button

Find people whose

Include disabled users in search

- contains
- doesn't contain
- is
- isn't

You'll be shown a list of matching users. If not, try again.

### Modifying a user

To change any user fields we just described -- password, contact info, rights, etc., first find the user you need (See *Searching for a user* subsection above). Then, assuming you get results:

click the ID of the correct user -> change the necessary fields in the Modify the user page -> click the Submit button

### Granting user rights

Before you grant a user any rights, you must **make her a staff user**:

click Configuration -> click Users -> click name of user (or search for a user, then click) -> check the Let this user be granted rights box -> click the Submit button at the bottom of the page

To grant group rights to a user:

click Configuration -> click Groups -> click a group name -> click User Rights in the left side navigation bar -> locate the correct user and select right(s) from the New rights listbox next to that user -> click the Submit button at the bottom of the page

Users granted rights this way will be able to create and modify this particular group (if granted the Admin Group right), add or remove members from this group (if granted the Admin Group Membership right), or add or remove themselves from this group (if granted the Modify Own Membership right).

See also the *Granting group rights* subsection of the *Groups* section to grant these same group rights to a subgroup or system group.

**To grant queue rights to a user:**

click Configuration -> click Queues -> click the name of a queue -> click User Rights in the left side navigation bar -> select right(s) from the New rights listbox next to the correct user(s) -> click the Submit button at the bottom of the page

Users granted rights this way will be able to use the rights in this queue only.

See also the *Granting queue rights* subsection of the *Queues* section.

**To grant global rights to a user:**

click Configuration -> click Global -> click User Rights in the left side navigation bar-> select right(s) from the New rights listbox next to the correct user(s) -> click the Submit button at the bottom of the page

Users granted rights this way will be able to use those rights in any queue.

**Revoking user rights**

**To make a user a non-staff user:**

click Configuration -> click Users ->click name of user (or search for a user, then click) -> clear the Let this user be granted rights checkbox -> click the Submit button at the bottom of the page

**To take queue rights away from a user:**

click Configuration -> click Queues -> click the name of the queue -> click User Rights in the left side navigation bar -> make sure you're looking at the correct user -> check the checkbox next to the right you want to revoke -> click the Submit button at the bottom of the page

**To take global rights away from a user:**

click Configuration -> click Global -> click User Rights -> make sure you're looking at the correct user -> check the checkbox next to the right you want to revoke -> click the Submit button at the bottom of the page

**Disabling a user**

You can't delete a user in RT, you can only disable him. That's because each user has data attached to him, so it's important for RT to keep some sort of record. The method for locking the user out of the system is disabling him. A disabled user won't be able to do anything and, by default, won't come up in user searches.

**To disable a user:**

click Configuration -> click Users -> click on a user's name, or search for a user as described in the *Searching for a user* subsection and then click the user's name -> clear the Let this user access RT checkbox -> click the Submit button

Now if someone searches for that user, he won't turn up unless the searcher checks the Include disabled users in search box on the search page.

**Re-enabling a user**

You may need to allow a disabled user back into the system.

**To re-enable a user:**

click Configuration -> click Users -> check the Include disabled users in search checkbox -> click the Go! button -> click the name of the disabled user you'd like to re-enable -> check the Let this user be granted rights checkbox -> click the Submit button

## Groups

A group is a set of users. Generally, groups will contain users with something in common. There are five types of groups:

**user-defined group:** a group created by a staff user of RT.

**system group:** a group across RT such as "staff" (all staff users), "non-staff" (all non-staff users), and "everyone" (every single user). You can't change the membership of system groups.

**role group:** a group related to a ticket or queue. Queues have the role groups "Cc" and "AdminCc," while tickets have Cc and AdminCc, plus "Requestor."

**personal group:** a group set up by a staff user for a purpose like delegating rights to it. As personal groups are mainly used for rights delegation, we've put them in the *Delegating rights* section of this chapter instead of the *Groups* section.

**subgroup:** a kind of user-defined group that's part of another group, generally to illustrate a hierarchy. A subgroup has the same rights as its group, but a group doesn't necessarily have the same rights as its subgroups. Let's say you had a group called "United States." It might have subgroups like "Massachusetts," "New York," and so on. Massachusetts could have a subgroup "Boston." You could give a right to United States (and Massachusetts and Boston would get the right), or give a right to Massachusetts (and Boston would also get the right, but United States does not), or to Boston (neither Massachusetts nor United States get the right).

### Creating a group

To **make a new group**:

click `Configuration` -> click `Groups` -> click `New Group` in the left side navigation bar -> type a name for the group in the `Name` text box -> if possible, type a little about the group in the `Description` text box -> click the `Submit` button at the bottom of the page

You'll have to take these basic creation steps before you can alter other aspects of the group, as described in the subsections below.

### Changing a group's name or description

This one, at least, is pretty easy:

click `Configuration` -> click `Groups` -> click a group name -> type new text in the `Name` and/or `Description` boxes -> click the `Submit` button at the bottom of the page

### Adding group members

There are two types of members you can add or remove from a group: users and groups. If you make a group a member of another group, it is then a subgroup.

To **add a user or subgroup** to a group:

click `Configuration` -> click `Groups` -> click the name of a group -> click `Members` in the left side navigation bar -> in the `Add members` listbox, select the name of the user or group you'd like to add -> click the `Submit` button

The `Add members` listbox is multiselect, so you can select and add several members at once.

### Removing group members

To **remove a user or subgroup** from a group:

click `Configuration` -> click `Groups` -> click the name of a group -> click `Members` in the left side navigation bar -> under `Current members`, check the box next to the user(s) or subgroup(s) you'd like to remove -> click the `Submit` button

You can check as many boxes as needed.

It is also possible to add new members while deleting other members; just do both before clicking `Submit`.

### Granting group rights

Group rights are slightly more complicated than user rights, but there are a few points that will help you keep them straight:

- Rights granted to a user-defined group belong to all members of that group, as long as they stay members.
- System groups: rights granted to the system group "everyone" go to all users, rights of "staff users" to go all staff users, and rights of "non-staff users" go to all non-staff users.
- Rights granted to role groups (requestor, owner, Cc, AdminCc) are granted to users based on their role for the current ticket.
- Groups can have the same rights as users, both for specific queues and globally.
- There are three rights (Admin Group, Admin Group Membership, Modify Own Membership) that apply to groups rather than queues.

#### To grant group rights to a user:

click Configuration -> click Groups -> click the name of a group -> click User Rights in the left side navigation bar -> select right(s) from the New rights listbox next to the correct user(s) -> click the Submit button

Users granted rights this way will be able to create and modify this particular group (if granted the Admin Group right), add or remove members from this group (if granted the Admin Group Membership right), or add or remove themselves from this group (if granted the Modify Own Membership right).

#### To grant group rights to a system group or subgroup:

click Configuration -> click Groups -> click the name of the group -> click Group Rights -> select right(s) from the New rights listbox next to the correct group(s) -> click the Submit button at the bottom of the page

Subgroups and system groups granted rights this way will be able to create and modify this particular group (if granted the Admin Group right), add or remove members from this group (if granted the Admin Group Membership right), or add or remove themselves from this group (if granted the Modify Own Membership right).

#### To grant queue rights to a group:

click Configuration -> click Queues -> click the name of the queue -> click Group Rights in the left side navigation bar -> select right(s) from the New rights listbox next to the correct group(s) -> click the Submit button at the bottom of the page

Groups granted rights this way will be able to use the rights in this queue only.

#### To grant global rights to a group:

click Configuration -> click Global -> click Group Rights -> select right(s) from the New rights listbox next to the correct group(s) -> click the Submit button at the bottom of the page

Groups granted rights this way will be able to use the rights in every queue.

### Revoking group rights

#### To take group rights away from a user:

click Configuration -> click Groups -> click the name of a group -> click User Rights in the left side navigation bar -> make sure you're looking at the correct user -> check the checkbox next to the right you want to revoke -> click the Submit button at the bottom of the page

#### To take group rights away from a system group or subgroup:

click Configuration -> click Groups -> click the name of the group -> click Group Rights in the left side navigation bar -> make sure you're looking at the correct group -> check the checkbox next to the right you want to revoke -> click the Submit button at the bottom of the page

#### To take queue rights away from a group:

click Configuration -> click Queues -> click the name of the queue -> click Group Rights in the left side navigation bar -> make sure you're looking at the correct group -> check the checkbox next to

the right you want to revoke -> click the Submit button at the bottom of the page

To **take global rights away from a group**:

click Configuration -> click Global -> click Group Rights -> make sure you're looking at the correct group -> check the checkbox next to the right you want to revoke -> click the Submit button at the bottom of the page

## Queues

For administrative purposes, it's worth noting that queues should be used to group together tickets with something in common. Also, queues can't be deleted, only disabled, so create with care. For instance, don't make a new queue for a one-off task or very short term project.

### Creating a queue

Remember that queues cannot be deleted, just disabled. But if you really need a new one:

click Configuration -> click Queues -> click New queue in the left side navigation bar

You'll be presented with some boxes to fill in (The only field you really need to fill in is Queue Name, but providing as much information as possible always helps):

**Queue Name:** usually a single word

**Description:** the purpose of the queue

**Reply Address:** the email address for the queue

**Comment Address:** an email address that receives internal comments

**Priority starts at:** This can be a value from 0-99. When a ticket is created in this queue, it will be given this priority by default. This is most useful when you have staff who work on more than one queue.

**Requests should be due in \_\_\_ days:** provide a number here and, when a ticket is created, it will be given a due date that number of days later.

**Over Time, priority moves toward:** Again, a value from 0-99. This is intended to provide a ceiling for a ticket's priority.

There is also a checkbox, checked by default, that says Enabled (Unchecking this box will disable this queue). Leave that alone unless you want to get rid of the queue (see also the *Disabling a queue* subsection of the *Queues* section).

When you're done, click the Submit button to create the queue. Then you can add or change information using the instructions in the sections *Modifying a queue*, *Adding watchers to a queue*, etc.

### Setting priority

RT user Luke Hankins has come up with a good system for using RT's priority system to manage a group of sysadmins. We've included it here so you can adapt it to your organization:

There are some tickets that will never get done. This is not a reflection on the aptitude of the sysadmins, it's just that we'll never have enough time to do all the little things. As long as tasks are given the right priority, the right things will get done in the right order.

We can identify five levels of priority for tasks:

- Low - Minor annoyance; there is a workaround.

Medium - Some annoyance; there is a workaround.

High - Lots of annoyance; there is a workaround.

Critical - Something is partly broken or someone can't do part of their job as a result. No workaround.

Fatal - Something is very broken or someone can't do any of their job. No workaround.

RT allows you to assign a priority from 0-99 to a ticket. We decided to assign the following ranges for each priority level: 1-10 for Low tickets, 11-20 for Medium tickets and so on to 41-50 for Fatal tickets. They start at the lowest

number in the range and slowly grow to the max.

We decided there will always be one person 'on-call' from our team, leaving the rest of the team to carry on with their work uninterrupted.

The on-call person is responsible for taking incoming tickets and giving them a priority, as well as setting the due dates, merging them with any existing tickets, etc. This must be done within a set time. I've used 8 business hours to good effect.

The on-call person is also required to take a crack at all Critical and Fatal tickets that come in on his watch. If it's not within his ken, it should be handed off to an able person.

The on-call person is *not* responsible for any Low/Medium/High tickets that come in after they have been given a priority. By definition, these tickets have a workaround and thus they can wait.

Everyone else is responsible for working their tickets in priority order and grabbing (or being given by management) more tickets when their queues get low.

(Thanks, Luke)

### **Finding a queue**

To **see a list of active queues**:

click `Configuration` -> click `Queues`

Under `Enabled Queues`, you'll see a list of just that. Click the name of any of these queues to modify it.

To **find disabled queues**:

click `Configuration` -> click `Queues` -> check the `Include disabled queues` in listing box -> click the `Go!` button

Now you'll see a list of `All Queues`, including disabled queues. Click the name of any of these queues to modify it.

### **Modifying a queue**

To **change a queue's basic information**:

click `Configuration` -> click `Queues` -> click the name of the queue you'd like to modify

Now you can change the queue's name, description, reply and comment addresses, priority, due date, and enabled/disabled status. Click the `Submit` button at the bottom of the page when you're done.

### **Adding watchers to a queue**

To **get to the right page** to add watchers to a queue:

click `Configuration` -> click `Queues` -> click the name of the queue you'd like to modify -> click `Watchers` in the left side navigation bar

Next, you'll need to **find the right user** by using the search boxes on that page and clicking the `Go!` button. (For instance, "Find people whose User ID is test.")

RT should return one or more matching usernames for your search (if not, try again. If you do have results, however, searching again will replace them with new results). Next to the username(s) of your choice:

select `Cc` or `AdminCc` from the drop-down menu under `Users` -> click the `Save Changes` button at the bottom of the page

### **Deleting watchers from a queue**

To **remove a user from Cc or AdminCc duty** on a queue:

click `Configuration` -> click `Queues` -> click the name of the queue you'd like to modify -> click `Watchers` in the left side navigation bar -> check the box(es) next to the user(s) you'd like to remove -> click the `Save Changes` button at the bottom of the page

## Scripts in queues

### Creating a scrip

#### To add a scrip to a queue:

click Configuration -> click Queues -> click the name of the queue -> click Scripts in the left side navigation bar-> click New scrip -> type a few relevant words in the Description box -> select a condition from the Condition drop-down menu -> select an action from the Action drop-down menu -> select a template from the Template drop-down menu -> click the Submit button

### Modifying a scrip

#### To modify a queue scrip

click Configuration -> click Queues -> click the name of the queue -> click Scripts in the left side navigation bar-> click the name of the scrip to modify -> make changes -> click the Submit button

You cannot modify scrips that apply to all queues.

### Deleting a scrip

#### To delete a queue scrip

click Configuration -> click Queues -> click the name of the queue -> click Scripts in the left side navigation bar-> check the box next to the scrip you'd like to delete -> click the Submit button

You cannot delete scrips that apply to all queues.

## Templates in queues

A template is essentially a form letter that goes along with a scrip action. A simple example could be, "Dear [name of requestor], your ticket has been created and its ticket ID is [this number]."

Templates allow access to RT's internals, so the right to create and edit templates should not be granted lightly.

### Creating a new scrip template

#### To create a new template:

click Configuration -> click Queues -> click the name of the queue -> click Templates in the left side navigation bar-> click New template in the left nav bar -> type a name, description, and content -> click the Submit button at the bottom of the page

**Name:** the name of the template, preferably something descriptive

**Description:** the words that will appear under the name of the template in RT's list of templates

**Content:** The form letter the users will see. This can include the subject line of the email at the top, just leave two lines between the header/subject and body/letter. Content can be simple or be personalized by including code like: This message has been automatically generated in response to the creation of a trouble ticket regarding: "{\$Ticket->Subject()}",

---

Note: The latest version of RT allows you to notify with attachments, rather than including the body of the message inline. Add a header to the top of the Content field of any template that reads

RT-Attach-Message: yes

Remember to leave two lines after the header.

---

### Modifying a scrip template

#### To modify a queue template:

click Configuration -> click Queues -> click the name of the queue -> click Templates in the left side navigation bar-> click the name of the template to modify -> make changes -> click the Submit button

### Deleting a scrip template

To **delete a queue template**:

click `Configuration` -> click `Queues` -> click the name of the queue -> click `Templates` in the left side navigation bar -> check the box next to the template you'd like to delete -> click the `Submit` button

## Custom fields in queues

### Adding a custom field

To **add a custom field for a queue**:

click `Configuration` -> click `Queues` -> click the name of the queue -> click `Custom Fields` in the left side navigation bar-> click `New custom field` in the left nav bar-> type the custom field's name and description in the `Name` and `Description` boxes -> select a format from the `Type` drop-down menu -> click the `Submit` button

Now that the custom field has been created, you'll **fill in its values**. These values are the same no matter what type of custom field you're creating. Not surprisingly, **Name** is the name of the custom field and **Description** is a short, optional description of what the custom field does.

**SortOrder** refers to the order in which a list of values appears. For example, assume you have values A, B, and C. If you leave `SortOrder` alone, the values will appear in the order A, B, C. If you give A the sort order of 2, B a 3, and C a 1, the values will appear in the order C, A, B.

Click the `Submit` button when you're done.

### Modifying a custom field

If you have the rights to handle tickets, you may **change a custom field's value for a ticket**. Select the new value from the ticket's `Basics` page .

You may also change a custom field's name, description, type, or value.

To **modify queue-specific custom fields** :

click `Configuration` -> click `Queues` -> click the name of the queue -> click `Custom Fields` in the left side navigation bar-> click the custom field to modify -> make changes -> click the `Submit` button

### Disabling a custom field

You can't delete a custom field, only disable it.

To **disable queue-specific custom fields** :

click `Configuration` -> click `Queues` -> click the name of the queue -> click `Custom Fields` in the left side navigation bar-> click the custom field to disable -> clear the `Enabled` checkbox -> click the `Submit` button

### Re-enabling a custom field

click `Configuration` -> click `Queues` -> click `Custom Fields` in the left side navigation bar-> check the `Include disabled custom fields in listing` checkbox -> click the `Go!` button -> click the name of the disabled custom field you'd like to re-enable -> check the `Enabled` checkbox -> click the `Submit` button

## Rights within a queue

To **grant queue rights to a group**:

click `Configuration` -> click `Queues` -> click the name of the queue -> click `Group Rights` in the left side navigation bar -> select right(s) from the `New rights` listbox next to the correct group(s) -> click the `Submit` button at the bottom of the page

The selected group(s) can now use this right in this queue only.

**To grant queue rights to a user:**

click Configuration -> click Queues -> click the name of the queue -> click User Rights in the left side navigation bar -> select right(s) from the New rights listbox next to the correct user(s) -> click the Submit button at the bottom of the page

The selected user(s) can now use this right in this queue only.

**Revoking queue rights**

**To take a queue right away from a user:**

click Configuration -> click Queues -> click the name of the queue -> click User Rights in the left side navigation bar -> make sure you're looking at the correct user -> check the checkbox next to the right you want to revoke -> click the Submit button at the bottom of the page

**To take a queue right away from a group:**

click Configuration -> click Queues -> click the name of the queue -> click Group Rights in the left side navigation bar -> make sure you're looking at the correct group -> check the checkbox next to the right you want to revoke -> click the Submit button at the bottom of the page

**Disabling a queue**

RT is a packrat, so you can never delete a queue. But you essentially put it out of commission by disabling it:

click Configuration -> click Queues -> click the name of the queue to be disabled -> clear the Enabled checkbox -> click the Submit button at the bottom of the page

**Re-enabling a queue**

To reactivate a disabled queue:

click Configuration -> click Queues -> check the Include disabled queues in listing checkbox -> click the Go! button -> click the name of the queue to re-enable -> check the Enabled checkbox -> click the Submit button

## Global

Change something globally and it changes in every queue. Not for the faint-hearted.

## Global scrips

Note that you shouldn't create a global scrip using the "On Create" condition and the "Create Ticket" scrip action, because it will create a loop.

**Adding a scrip**

We'll start by adding a new scrip made of existing conditions, actions, and templates. For now, we're going to skip the scrip fields Custom condition, Custom action preparation code, and Custom action cleanup code.

**To add a scrip globally:**

click Configuration -> click Global -> click Scrips -> click New scrip in the left side navigation bar -> type a few relevant words in the Description box -> select a condition from the Condition drop-down menu -> select an action from the Action drop-down menu -> select a template from the Template drop-down menu -> click the Submit button

**Modifying a scrip**

**To modify a global scrip**

click Configuration -> click Global -> click Scrips -> click the name of the scrip to modify -> make changes -> click the Submit button

## Deleting a scrip

### To delete a global scrip

click Configuration -> click Global -> click Scripts -> check the box next to the scrip you'd like to delete -> click the Submit button

## Global templates

A template is essentially a form letter that goes along with a scrip action. A simple example could be, "Dear [name of requestor], your ticket has been created and its ticket ID is [this number]."

Templates allow access to RT's internals, so the right to create and edit templates should not be granted lightly.

### Creating a scrip template

#### To create a new template:

click Configuration -> click Global -> click Templates -> click New template in the left side navigation bar -> type a name, description, and content -> click the Submit button at the bottom of the page

**Name:** the name of the template, preferably something descriptive

**Description:** the words that will appear under the name of the template in RT's list of templates

**Content:** The form letter the users will see. This can include the subject line of the email at the top, just leave two lines between the header/subject and body/letter. Content can be simple or be personalized by including code like: This message has been automatically generated in response to the creation of a trouble ticket regarding: "`{ $Ticket->Subject() }`",

---

Note: The latest version of RT allows you to notify with attachments, rather than including the body of the message inline. Add a header to the top of the Content field of any template that reads

RT-Attach-Message: yes

Remember to leave two lines after the header.

---

### Modifying a scrip template

#### To modify a global template:

click Configuration -> click Global -> click Templates -> click the name of the template to modify -> make changes -> click the Submit button

### Deleting a scrip template

#### To delete a global template:

click Configuration -> click Global -> click Templates -> check the box next to the template you'd like to delete -> click the Submit button

## Global custom fields

### To add a custom field globally:

click Configuration -> click Global -> click Custom Fields -> click New custom field in the left side navigation bar -> type the custom field's name and description in the Name and Description boxes -> select a format from the Type drop-down menu -> click the Submit button

Now that the custom field has been created, you'll **fill in its values**. These values are the same no matter what type of custom field you're creating.

Not surprisingly, **Name** is the name of the custom field and **Description** is a short, optional description of what the custom field does.

**SortOrder** refers to the order in which a list of values appears. For example, assume you have values A, B, and C. If

## DRAFT, April 13, 2003

you leave SortOrder alone, the values will appear in the order A, B, C. If you give A the sort order of 2, B a 3, and C a 1, the values will appear in the order C, A, B.

Click the Submit button when you're done.

If you have the rights to handle tickets, you may **change a custom field's value for a ticket**. Select the new value from the ticket's Basics page .

You may also change a custom field's name, description, type, or value.

To **modify global custom fields** :

click Configuration -> click Global -> click Custom Fields -> click the custom field to modify  
-> make changes -> click the Submit button

To **disable global custom fields** :

click Configuration -> click Global -> click Custom Fields -> click the custom field to disable  
-> clear the Enabled checkbox -> click the Submit button

To **re-enable a global custom field**

click Configuration -> click Global -> click Custom Fields -> check the Include disabled custom fields in listing checkbox -> click the Go! button -> click the name of the disabled custom field you'd like to re-enable -> check the Enabled checkbox -> click the Submit button

## Global group Rights

To **grant global rights to a group**:

click Configuration -> click Global -> click Group Rights -> select right(s) from the New rights listbox next to the correct group(s) -> click the Submit button at the bottom of the page

To **take global rights away from a group**:

click Configuration -> click Global -> click Group Rights -> make sure you're looking at the correct group -> check the checkbox next to the right you want to revoke -> click the Submit button at the bottom of the page

## Global user Rights

To **grant global rights to a user**:

click Configuration -> click Global -> click User Rights -> select right(s) from the New rights listbox next to the correct user(s) -> click the Submit button at the bottom of the page

To **take global rights away from a user**:

click Configuration -> click Global -> click User Rights -> Rights -> make sure you're looking at the correct user -> check the checkbox next to the right you want to revoke -> click the Submit button at the bottom of the page

**DRAFT, April 13, 2003**

# Appendix 1

## Glossary

This glossary covers RT terms, both the ones you'll find in version 3 and outdated terms from earlier versions. Outdated terms won't have definitions, but you'll be referred to the current term. Italics indicate a word that is defined elsewhere in the glossary.

Some of these definitions are a bit circular -- this is intentional, so you can see how concepts in RT relate to each other, but may be confusing for beginners. If you are new to RT, please read the *Chapter 1: Introduction* to understand RT's basic concepts and terminology.

**ACL:** Access Control List; a scheme that supports and keeps track of *rights*.

**action:** see *scrip action*, the preferred term.

**AdminCc:** a *Cc* that also gets copies of *comments* and generally are allowed to work on the *ticket*.

**Cc:** someone who should get copies of any *replies* that go to the *requestor*.

**child:** *relationship* indicating a smaller, narrower subproject of a larger, broader project (the *parent*).

**comment:** a private note for *staff* not visible to the *requestor*.

**condition:** see *scrip condition*, the preferred term.

**correspondence:** old term no longer used in RT 3; see *reply*.

**custom field:** database fields that can be customized to each organization's needs.

**dead:** old term no longer used in RT 3; see *deleted*.

**deleted:** *status* indicating that a *ticket* should not have been in the system and is being removed.

**depended on by:** *relationship* indicating that this *ticket* must be *resolved* before another ticket can be resolved.

**depends on:** *relationship* indicating that this *ticket* can't be *resolved* unless another ticket is resolved first.

**due date:** the date by which the *ticket* should be *resolved*. See also *final priority*.

**final priority:** the *priority* a *ticket* moves towards as it approaches its *due date*.

**global:** across the entire system.

**group:** a set of related *users*; useful when assigning *rights*.

**history:** a permanent record of what has happened to the *ticket* since (and including) its creation.

**interested party:** old term no longer used in RT 3; see *watcher*.

**keyword:** old term no longer used in RT 3; see *custom field*.

**link:** see *relationship*.

**member:** a *user* or *subgroup* that is part of a *group*.

**new:** *status* indicating that a *ticket* has just been created and has not been touched yet.

**non-staff user:** a *user* who usually has no *rights* to do things in RT; often a *requestor* whose account was automatically created when the request was sent.

**open:** *status* indicating that a *ticket* is getting worked on.

**owner:** the person responsible for a *ticket* and its resolution.

**parent:** *relationship* indicating a large umbrella project containing subprojects (*children*).

**permission:** old term no longer used in RT 3; see *right*.

**people:** a ticket's *watchers* and *owner*.

**personal group:** a *group* set up by a *staff user* for a purpose like delegating *rights* to it.

**priority:** how important a *ticket* is, represented as numerical scale from 0-99, with 99 being the highest priority. See also *final priority*.

**privilege:** old term no longer used in RT 3; see *right*.

**privileged user:** old term no longer used in RT 3; see *staff user*.

**pseudogroup:** old term no longer used in RT 3; see *system group*.

**queue:** the central administrative domain of RT. A line of *tickets* waiting to be worked on, but it's also, to some extent, a ticket's category.

**referred to by:** *relationship* indicating that this *ticket* is relevant to another ticket; *watchers* of the second ticket should consult the ticket for context.

**refers to:** *relationship* indicating that another *ticket* is relevant to this one; *watchers* should consult the second ticket for context.

**rejected:** *status* indicating that the *ticket* will not ever be *resolved* but will remain recorded in the system.

**relationship:** one *ticket*'s connection to another; either a loose association or a dependency. See also *depends on*, *depended on by*, *refers to*, *referred to by*, *parent*, and *child*.

**reply:** a public remark that a *requestor* can see.

**request:** see *ticket*, the preferred term.

**requestor:** the person who asked for something to get done; the *ticket's* reason for being.

**resolved:** *status* indicating that work on a *ticket* has been completed.

**right:** ability given to a *user* or *group* to do something in RT. Rights can be granted by *queue* or for the entire system. Rights can be given to *groups*, *roles*, or individual *users*.

**role:** type of *watcher*, such as *owner*, *requestor*, *Cc*, and *AdminCC*.

**scrip:** a custom notification that will automatically take a certain *scrip action* in response to a certain *scrip condition*, often using a *template*.

**scrip action:** something a *scrip* does in response to a *scrip condition*.

**scrip condition:** something that has to happen to activate a *scrip*, like a *ticket* being created or a *comment* being made.

**staff user:** a *user* who can be granted *rights*. See also *non-staff user*.

**stalled:** *status* indicating that a *ticket* cannot be worked on for now, but will open again when someone adds a *comment* or *reply*.

**status:** classification that shows a *ticket's* progress through the system. See also *new*, *open*, *stalled*, *resolved*, *rejected*, and *deleted*.

**subgroup:** a group that's part of another group, generally to illustrate a hierarchy.

**system group:** a *group* across RT such as "staff" (all *staff users*), "non-staff" (all *non-staff users*), and "everyone" (every *user*). You can't change the membership of system groups.

**template:** essentially an email form letter that goes along with a *scrip action* in a *scrip*.

**ticket:** the central object in RT, the thing that needs to get done. Tickets have metadata attached to them such as *owner*, *status*, *queue*, etc.

**ticket stub:** email message received by a *requestor* to confirm that his request has been received and a ticket created.

**unprivileged user:** old term no longer used in RT 3; see *non-staff user*.

**updates:** see *comment* and *reply*, the more specific terms for updates..

**user:** anyone who interacts with RT in any way. See *staff* and *non-staff*.

**watcher:** someone who is interested in a *ticket*. See also *owner*, *requestor*, *Cc*, and *AdminCc*.

**DRAFT, April 13, 2003**

# Appendix 2

## Rights

This is a reference for rights in RT. The rights are broken down into three tables: global, queue, and group. Within those tables, rights are listed alphabetically.

**Table 1: Global rights**

name of right	what users with this right can do
Admin All Personal Groups	change name, description, and members of any personal group
Admin Custom Fields	add, delete, and modify custom fields
Admin Group	create groups; change groups' name, description, and rights
Admin Group Membership	add and remove members from a group
Admin Own Personal Groups	change name, description, and members of their own personal groups
Admin Queue	create, enable, and dis able queues; change queues' name, description, reply and comment address, priority, and due date
Admin Users	create users; change anything on users' Basics screen: username, email, name, nickname, Unix login, address, phone numbers, comments about the user, signature, password, and rights.
Comment On Ticket	add comments to tickets
Create Ticket	create tickets
Delegate Rights	assign their own rights to other users
Delete Ticket	delete tickets
Modify ACL	change users' and groups' rights. Users with this right should also have the right Show ACL.
Modify Own Membership	add or remove themselves from a group

**DRAFT, April 13, 2003**

Modify Queue Watchers	add and remove Ccs and AdminCcs
Modify Scrips	add, delete, and change the content of scrips
Modify Self	modify his own email address, real name, nickname, extra contact information, password, organization, address, phone numbers, and signature.
Modify Template	create, modify, and delete scrip templates
Modify Ticket	change tickets' subject, status, time worked, time left, priorities, queue, dates, watchers, owner, and relationships to other tickets; add comments and replies to tickets
Own Ticket	own (be responsible for) a ticket
Reply To Ticket	add replies to a ticket
See Queue	know that a queue exists; queues will appear in visible listings for this user
Show ACL	see the rights granted to users and groups
Show Scrips	see scrips
Show Template	see a scrip's email template
Show Ticket	see tickets
Show Ticket Comments	see comments attached to tickets
Super User	everything; this user has all possible rights within RT
Watch	sign up as a requestor or Cc
Watch As AdminCc	sign up as an AdminCC

**Table 2: Queue rights**

<b>name of right</b>	<b>what users with this right can do</b>
Admin Custom Fields	add, delete, and modify custom fields
Admin Queue	create, enable, and disable queues; change queues' name, description, reply and comment address, priority, and due date
Comment On Ticket	add comments to tickets
Create Ticket	create tickets
Delete Ticket	delete tickets
Modify ACL	change users' and groups' rights. Users with this right should also have the right Show ACL.
Modify Queue Watchers	add and remove Ccs and AdminCcs

Modify Scripts	add, delete, and change the content of scrips
Modify Template	create, modify, and delete scrip templates
Modify Ticket	change tickets' subject, status, time worked, time left, priorities, queue, dates, watchers, owner, and relationships to other tickets; add comments and replies to tickets
Own Ticket	own (be responsible for) a ticket
Reply To Ticket	add replies to a ticket
See Queue	know that a queue exists; queues will appear in visible listings for this user
Show ACL	see the rights granted to users and groups
Show Scrips	see scrips
Show Template	see a scrip's email template
Show Ticket	see tickets
Show Ticket Comments	see comments attached to tickets
Watch	sign up as a requestor or Cc
Watch As AdminCc	sign up as an AdminCC

### Table 3: Group rights

name of right	what users with this right can do
Admin Group	create groups; change groups' name, description, and rights
Admin Group Membership	add and remove members from a group
Modify ACL	change users' and groups' rights. Users with this right should also have the right Show ACL.
Modify Own Membership	add or remove themselves from a group

**DRAFT, April 13, 2003**

# Appendix 3

## Scripts

These tables explain what scrip conditions, actions, and templates do. If you're still unclear about scrips, you can read a condition as the first part of a sentence, an action as the second part, and the template as the third part:

When <condition>, <action> with <template>.

**Table 1: Conditions**

condition name	what it means	possible template
On Create	When a new ticket is created	Global template: Autoreply
On Transaction	When a ticket is updated or changed in any way	Global template: Transaction
On Reply	When a watcher replies to a ticket	Global template: Reply Global template: Admin Reply
On Comment	When a watcher comments on a ticket (privately)	Global template: Admin Comment
On Correspond	When a watcher corresponds to (replies to) a ticket (publicly)	Global template: Blank
On Status	When the ticket's status changes	Global template: Status Changed
On Owner Change	When the ticket gets a new owner	Global template: Transaction
On Queue Change	When the ticket's is put in a different queue	Global template: Transaction
On Resolve	When the ticket is resolved (completed)	Global template: Resolved

Note: don't create a global scrip using the "On Create" condition and the "Create Ticket" scrip action. This will create a loop.

## Table 2: Actions

action name	what it means
Auto Reply to Requestors	email an Auto Reply template to the ticket's requestor
Notify Requestors	email the ticket's requestor
Notify Owner As Comment	email the ticket's owner. "As Comment" means the return address of the email is RT's comment address and all responses will be posted as comments not visible to the requestor.
Notify Owner	email the ticket's owner.
Notify Admin Ccs As Comment	email the ticket's Admin Ccs. "As Comment" means the return address of the email is RT's comment address and all responses will be posted as comments not visible to the requestor.
Notify Admin Ccs	email the ticket's Admin Ccs.
Notify Requestors And Ccs As Comment	email the ticket's requestor and Ccs. "As Comment" means the return address of the email is RT's comment address and all responses will be posted as comments not visible to the requestor.
Notify Requestors And Ccs	email the ticket's requestor and Ccs.
Notify All Watchers As Comment	email the ticket's requestor, owner, Ccs, and Admin Ccs. "As Comment" means the return address of the email is RT's comment address and all responses will be posted as comments not visible to the requestor.
Notify All Watchers	email the ticket's requestor, owner, Ccs, and Admin Ccs.
Notify Other Recipients As Comment	email the addresses in the Cc: and Bcc: fields of a ticket's reply or comment. "As Comment" means the return address of the email is RT's comment address and all responses will be posted as comments not visible to the requestor.
Notify Other Recipients	email the addresses in the Cc: and Bcc: fields of a ticket's reply or comment
Open Tickets	automatically opens a stalled, resolved, or rejected ticket when anyone replies to it; automatically opens new tickets when anyone except the requestor replies to it. Does not apply to tickets that are already open.

## Table 3: Templates

template name	what the user sees
Global template: Autoreply	A form letter saying a ticket has been created and what the ticket's ID is.

**DRAFT, April 13, 2003**

Global template: Transaction	A form letter saying a ticket has been updated or changed in some way.
Global template: Admin Reply	A form letter saying a watcher sent a message and what the message was. Includes a URL linking to the message in RT.
Global template: Reply	A form letter saying a watcher sent a message and what the message was.
Global template: Admin Comment	A form letter saying a watcher sent a message and what the message was. Includes a URL linking to the message in RT and a note that says the requestor cannot see the message.
Global template: Status Changed	A form letter saying a ticket's status (open, new, stalled, etc.) has been changed.
Global template: Resolved	A form letter saying a ticket has been resolved.

**DRAFT, April 13, 2003**

# Appendix 4

## Building Apache with mod\_perl

The mod\_perl guide can be found at: <http://perl.apache.org/docs/1.0/guide/>

The following is an excerpt, which should work for most people. Download the source tarballs for apache and for mod\_perl, and then:

```
% 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 DO_HTTPD=1
EVERYTHING=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/local/apache tree and mod\_perl into the site\_perl hierarchy of your existing Perl installation.

Do not omit the EVERYTHING=1 parameter.

**DRAFT, April 13, 2003**

# Appendix 5

## Creating Approvals

Approvals are a useful way to get tickets OK'd by management without creating a new system. A supervisor's approval can be a ticket; another ticket can depend on that ticket being resolved.

The custom programming to create tickets involves the Create Tickets scrip action and a template that asks a supervisor to look at the ticket asking for approval.

Here is an example of the process by RT's head honcho, Jesse Vincent:

### NAME

RT::Action::CreateTickets

Create one or more tickets according to an externally supplied template.

### SYNOPSIS

===Create-Ticket: codereview

Subject: Code review for {\$Tickets{"TOP"}->Subject}

Depended-On-By: {\$Tickets{"TOP"}->Id}

Content: Someone has created a ticket. you should review and approve it, so they can finish their work

ENDOFCONTENT

### DESCRIPTION

Using the "CreateTickets" ScripAction and mandatory dependencies, RT now has the ability to model complex workflow. When a ticket is created in a queue that has a "CreateTickets" scripaction, that ScripAction parses its "Template"

### FORMAT

CreateTickets uses the template as a template for an ordered set of tickets to create. The basic format is as follows:

===Create-Ticket: identifier

Param: Value

Param2: Value

Param3: Value

Content: Blah

blah

blah

ENDOFCONTENT

```
===Create-Ticket: id2
Param: Value
Content: Blah
ENDOFCONTENT
```

Each ===Create-Ticket: section is evaluated as its own Text::Template object, which means that you can embed snippets of perl inside the Text::Template using {} delimiters, but that such sections absolutely can not span a ===Create-Ticket boundary.

After each ticket is created, it's stuffed into a hash called %Tickets so as to be available during the creation of other tickets during the same ScripAction. The hash is prepopulated with the ticket which triggered the ScripAction as \$Tickets{'TOP'}; you can also access that ticket using the shorthand \$TOP.

A simple example:

```
===Create-Ticket: codereview
Subject: Code review for {$Tickets{'TOP'}->Subject}
Depended-On-By: {$Tickets{'TOP'}->Id}
Queue: Approvals
Type: Approval
Content: Someone has created a ticket. you should review and approve it, so they can finish their work
ENDOFCONTENT
```

A convoluted example:

```
===Create-Ticket: approval
{ # Find out who the administrators of the group called "HR"
  # of which the creator of this ticket is a member
  my $name = "HR";

  my $groups = RT::Groups->new($RT::SystemUser);
  $groups->LimitToUserDefinedGroups();
  $groups->Limit(FIELD => "Name", OPERATOR => "=", VALUE =>
"$name");
  $groups->WithMember($TransactionObj->CreatorObj->Id);

  my $groupid = $groups->First->Id;

  my $adminccs = RT::Users->new($RT::SystemUser);
  $adminccs->WhoHaveRight(
    Right => "AdminGroup",
    Object => $groups->First,
    IncludeSystemRights => undef,
    IncludeSuperusers => 0,
    IncludeSubgroupMembers => 0,
  );

  my @admins;
  while (my $admin = $adminccs->Next) {
    push (@admins, $admin->EmailAddress);
  }
}
Queue: Approvals
Type: Approval
AdminCc: {join ("\nAdminCc: ", @admins) }
Depended-On-By: {$Tickets{'TOP'}->Id}
```

## DRAFT, April 13, 2003

```
Refers-To: {$Tickets{"TOP"}->Id}
Subject: Approval for ticket: {$Tickets{"TOP"}->Id} - {$Tickets{"TOP"}->Subject}
Due: {time + 86400}
Content-Type: text/plain
Content: Your approval is requested for the ticket {$Tickets{"TOP"}->Id}: {$Tickets{"TOP"}->Subject}
Blah
Blah
ENDOFCONTENT
===Create-Ticket: two
Subject: Manager approval
Depended-On-By: {$Tickets{"TOP"}->Id}
Refers-On: {$Tickets{"approval"}->Id}
Queue: Approvals
Content-Type: text/plain
Content:
Your approval is required for this ticket, too.
ENDOFCONTENT
```

### ACCEPTABLE FIELDS

A complete list of acceptable fields for this beastie:

- \* Queue => Name or id# of a queue
- Subject => A text string
- ! Status => A valid status. defaults to 'new'
- Due => Dates can be specified in seconds since the epoch to be handled literally or in a semi-free textual format which RT will attempt to parse.
  
- Starts =>
- Started =>
- Resolved =>
- Owner => Username or id of an RT user who can and should own this ticket
- + Requestor => Email address
- + Cc => Email address
- + AdminCc => Email address
- TimeWorked =>
- TimeEstimated =>
- TimeLeft =>
- InitialPriority =>
- FinalPriority =>
- Type =>
- +! DependsOn =>
- +! DependedOnBy =>
- +! RefersTo =>
- +! ReferredToBy =>
- +! Members =>
- +! MemberOf =>
- Content => content. Can extend to multiple lines.  
Everything within a template after a Content: header is treated as content until we hit a line containing only ENDOFCONTENT
- ContentType => the content-type of the Content field
- CustomField-<id#> => custom field value

**DRAFT, April 13, 2003**

Fields marked with an \* are required. Fields marked with a + may have multiple values, simply by repeating the fieldname on a new line with an additional value.

Fields marked with a ! are postponed to be processed after all tickets in the same actions are created. Except for 'Status', those fields can also take a ticket name within the same action (i.e. the identifiers after ==Create-Ticket), instead of raw Ticket ID numbers.

When parsed, field names are converted to lowercase and have -s stripped. Refers-To, RefersTo, refersto, refers-to and r-e-f-e-r-s-tO will all be treated as the same thing.

**DRAFT, April 13, 2003**

# Appendix 6

## Perl Modules

These are the required perl modules in RT 3.0.0:

```
$deps{'CORE'} = [ _( << '.') ];
```

Digest::MD5

DBI 1.18

Test::Inline

Class::ReturnValue 0.40

DBIx::SearchBuilder 0.80

Text::Template

File::Spec 0.8

HTML::Entities

Net::Domain

Log::Dispatch 2.0

Locale::Maketext

Locale::Maketext::Lexicon 0.10

Locale::Maketext::Fuzzy

MIME::Entity 5.108

Mail::Mailer 1.57

Net::SMTP

Text::Wrapper

Time::ParseDate

File::Temp

Term::ReadKey

Text::Autoformat

Text::Quoted

.

**\$deps{'MASON'} = [ \_( << '.') ];**

Params::Validate 0.02

Cache::Cache

Exception::Class

HTML::Mason 1.16

MLDBM

Ermo

FreezeThaw

Digest::MD5

CGI::Cookie 1.20

Storable

Apache::Session 1.53

.

**\$deps{'MAILGATE'} = [ \_( << '.') ];**

HTML::TreeBuilder

HTML::FormatText

Getopt::Long

LWP::UserAgent

.

**\$deps{'CLI'} = [ \_( << '.') ];**

Getopt::Long 2.24

.

**\$deps{'DEV'} = [ \_( << '.') ];**

Regexp::Common

Time::HiRes

Test::Inline

WWW::Mechanize

.

**\$deps{'FASTCGI'} = [ \_( << '.') ];**

CGI

FCGI

CGI::Fast

.

**\$deps{'MODPERL1'} = [ \_( << '.') ];**

CGI

Apache::Request

Apache::DBI

.

**\$deps{'MODPERL2'} = [ \_( << '.') ];**

CGI 2.89

Apache::DBI

.

**\$deps{'I18N-COMPAT'} = [ \_( << '.') ];**

Text::Iconv

Encode::compat 0.04

.

**\$deps{'MYSQL'} = [ \_( << '.') ];**

DBD::mysql 2.0416

.

**\$deps{'POSTGRESQL'} = [ \_( << '.') ];**

DBD::Pg

.