Request Tracker Installation Guide

Abstract	:	Describes the Installation and upgrade of Request Tracker V3.
Issue	:	14
Date	:	08/02/2007

Document History

Issue	Author(s)	Date	Description Change
1	N. Metrowsky	02/03/2006	Document Creation
2	N Metrowsky	02/16/2006	Added section for installation of Clickable Hypertext Links in Tickets Callback/Patch
3	N Metrowsky	03/07/2006	Updated patches to the Statistics subsystem.
4	N Metrowsky	04/24/2006	Added information about modified templates. Removed extraneous material.
5	N Metrowsky	06/27/2006	Added section on performance tuning.
6	N Metrowsky	08/11/2006	Added information on displaying Stalled tickets in “25 highest priority tickets I own…” section on Request Tracker Home Page. Added a chapter which describes the Request Tracker upgrade procedure from V3.4.4 to V3.6.1.
7	N Metrowsky	10/11/2006	Added section describing a modification made to warn users about connectivity issues when relocating laptop computers.
8	N Metrowsky	12/12/2006	Added section describing quick linking to Queues.
9	N Metrowsky	01/03/2007	Revised information related to Request Tracker V3.6.1 to include Request Tracker V3.6.3
10	N Metrowsky	01/17/2007	Added more customization information
11	N Metrowsky	02/06/2007	Added information on Change of Ticket Ownership Tell Requestor scrip.
12	N Metrowsky	03/12/2007	Addition of patch which requires Time Worked to be populated, when a ticket is Resolved.
13	N Metrowsky	06/20/2007	Added patch to generate list of queues a user has access to on the main menu. This patch eliminates the need to modify the menu each time a queue is added or removed.
14	N Metrowsky	08/02/2007	Added additional database tuning parameters to improve Request Tracker performance. In summary, turned on database caching.

Table of Contents

1. Installation of Request Tracker

1.1. *Introduction*

From the Request Tracker website at http://www.bestpractical.com:

RT is an enterprise-grade ticketing system which enables a group of people to intelligently and efficiently manage tasks, issues, and requests submitted by a community of users.

The RT platform has been under development since 1996, and is used by systems administrators, customer support staffs, IT managers, developers and marketing departments at thousands of sites around the world.

Written in object-oriented Perl, RT is a high-level, portable, platform independent system that eases collaboration within organizations and makes it easy for them to take care of their customers.

RT manages key tasks such as the identification, prioritization, assignment, resolution and notification required by enterprise-critical applications including project management, help desk, NOC ticketing, CRM and software development.

1.2. *About this manual*

This manual is a collection of notes and summaries concerning the installation of Request Tracker, and its' related software modules. Therefore, this document is more than an installation guide, as it provides as complete picture, as possible, of how Request Tracker was installed at DigitalGlobe. For example, Chapter 2, and the Appendix are provided for historical purposes, but could become useful in the event a major upgrade installation is required in the future.

In addition, installation of Request Tracker FAQ Manual (RTFM) and RTx::AssetTracker are covered in more detail in there respective manuals. It is in the best interest of the installer to review all the documentation, as they have to consider all components of the Request Tracker before attempting an upgrade.

1.3. *Beginning the Installation*

The installation of Request Tracker is a complicated process, so care must be followed in order to install the package correctly. Request Tracker requires the installation of these software products, in the following order, at the version levels indicated:

1. Perl V5.8.5 or higher
2. MySQL V4.1.0 or higher
3. Apache V2.0.50 or higher
4. mod_perl V2.0.0 or higher
5. Request Tracker V3.4.4 or higher
6. RTx::ssetTracker V1.2.2 or higher
7. RTx::RightsMatrix V0.02.05 or higher
8. Request Tracker FAQ Manual (RTFM) V3.1.10 or higher

9. Request Tracker Statistics 3 V0.1.8 or higher

1.3.1. Cautions and Notes

All software installed, except for RPM files, will be installed in the **/usr/local** directory tree, with the exception of Request Tracker, which is installed in **/opt**. The purpose of doing this is to make it easy to remove the installed software, if the need arises. In addition, utilities like **rpm** and **up2date** can still be run on the Linux environment, without interfering with the software modules discussed in this document.

This document assumes that Request Tracker V3.4.4 or higher is being installed. Request Tracker distributions prior to V3.4.4 are only supported on Apache V1 series web servers. Also, in addition to installing Request Tracker, the RTx::AssetTracker extension to Request Tracker will also have to be installed.

Before starting, make sure there are enough system and disk space resources for the installation. A Linux PC with a Pentium IV CPU, 1 Gb of memory and 80 Gb of disk space should handle the Request Tracker environment more than sufficiently.

A warning before starting, on many RedHat and Fedora Linux systems, Perl has been pre-installed. There are some applications, e.g. **redhat-config-printer** or **system-config-printer**, which requires the preinstalled Perl to function properly. Therefore, it is best to use a dedicated system for Request Tracker, to avoid any dependency issues.

Another warning, Redhat and Fedora Linux systems also include an installation of MySQL. Unfortunately, on RedHat Enterprise V3 installations it is not new enough for Request Tracker. The preinstalled MySQL should be removed, as follows:

```
rpm -qa | grep -i mysql
rpm -e <the list of rpm set returned via the previous command>
```

As noted with Perl above, there may be applications that depend on the preinstalled MySQL. So, it is best to use RedHat Enterpirse WS as the base installation, as it comes only with a MySQL client preinstalled. If RedHat Enterprise AS or ES is used as a target system, then several products that depend on the preinstalled MySQL will have to be removed, as these systems provide a MySQL server and a MySQL client.

Finally, for the sanity of the installer, it is best to install all the devel RPMs on the target Linux system, except for MySQL libpng and Perl. If not, the installer will be forced to install one devel RPM after another, as they proceed through this documentation. It is possible that the installed Linux already has the devel RPMs installed, to check:

```
rpm -qa | grep devel
```

1.4. Installation Procedure

1.4.1. Install Perl

Request Tracker is written Perl, so a Perl interpreter needs to be installed on the target system. Perl V5.8.5 or higher is required for the installation of Request Tracker V3.4.4; it will not install, unless the correct Perl version is installed.

Download Perl from http://www.perl.org, and issue the following commands:

```
tar -xzf perl-5.8,7.tar.gz
cd perl-5.8.7
./Configure (take the default answers)
make
make test
make install
```

Perl should now be installed in the **/usr/local** hierarchy.

Important: Read the notice in the **Cautions and Notes** section concerning installation of Perl on RedHat and Fedora systems.

Now, deactivate the preinstalled Perl. The deactivation is done by renaming the existing Perl image and creating a soft link to the new Perl location

```
mv /usr/bin/perl /usr/bin/perl.old
ln -s /usr/local/bin/perl perl
```

Warning: The old Perl can be reactivated, if needed; however, restoring the old Perl on a running Request Tracker system will cause unexpected results and is strongly discouraged.

1.4.2. Install MySQL

Request Tracker requires a database engine to store various fields and other information; therefore, MySQL V4.1.0 or higher be installed on the target system.

Important: Before installing MySQL, please read the note in the **Cautions and Notes** section, concerning installation of MySQL on RedHat and Fedora systems.

Download MySQL from http://www.mysql.org, and issue the following commands:

```
tar -xzf mysql-4.1.15
cd mysql-4.1.5
./configure –prefix=/usr/local/mysql --with-innodb –enable-thread-safe-client \
--enable-ssl
make
make test
make install
```

Make sure the user name **mysql** and group **mysql** exist, else issue the following commands:

```
groupadd mysql
useradd –g mysql mysq
```

Next, create a copy of **/usr/loca/etc/my.cnf:**

```
mkdir /usr/local/mysql/etc
cp support-files/my-medium.cnf /usr/local/mysql/etc/my.cnf
```

Edit the my.cnf file as follows:

Change: **max_allowed_packet** from `1M` to `64M`
Add: `connect_timeout = 120` after **max_allowed_packet**

Note: Please see the chapter on Performance Tuning for more information.

Next, install the initial databases:

```
mkdir /usr/local/mysql/var
mkdir /usr/local/mysql/log
cd /usr/local/mysql
bin/mysql_install_db –user=mysql
chown –R mysql:mysql /usr/local/mysql/var
```

Next, create **/etc/init.d/mysqld**, a customized copy is provided in the Appendix. **mysqld** is the startup rc script for MySQL. The customization entails making sure the file PATHs are correct throughout the script. Install the attached copy as **/etc/init.d/mysqld**.

Next, create **/usr/local/mysql/sbin/safe_mysqld**, a customized copy is provided in the Appendix. **safe_mysql** is a start up script for the MySQL software product. The customization entails making sure the file PATHs are correct throughout the script. Install the attached copy as **/usr/local/mysql/bin/safe_mysqld**.

Insert the following command in **/etc/rc.local**, in order to start the MySQL database server when the system is rebooted:

```
/etc/init.d/mysqld start
```

At this point, the new MySQL can be started via the command:

```
/etc/init.d/mysqld start
```

Finally, create a link to run the MYSQL Administration script:

```
cd /usr/local/bin
ln –s /usr/local/mysql/bin/mysql
```

1.4.3. Install Apache

Request Tracker needs a web server to properly display its' web pages to the user. Request Tracker V3.4.4 requires the installation of Apache V2.0.50 or higher, for it to function properly.

Download Apache from http://www.apache.org, and issue the following commands:

```
tar -xzf httpd-2.0.55.tar.gz
cd httpd-2.0.55
/configure --prefix=/usr/local/apache2 --enable-so --enable-ssl \
--enable-cgi --enable-rewrite
make
make install
```

Create the following group and user, if they do not already exist:

```
groupadd apache
useradd -g apache -d /localhost/home/apache apache
```

Just in case **Apache** complains about a missing **httpd.conf** file, issue the following commands

```
mkdir /usr/local/conf
cd /usr/local/conf
ln -s /usr/local/apache2/conf/httpd.conf
```

To prepare Apache for Request Tracker, modify **/usr/local/apache2/conf/httpd.conf**, as follows:

Insert:

```
Listen 80
Listen 81
```

Set up virtual hosts for Request Tracker:

```
<VirtualHost 10.10.36.95:81>
   ServerName helpdesk:81.digitalglobe.com
   DocumentRoot /usr/local/apache2/htdocs
</VirtualHost>

<VirtualHost 10.10.36.95>
   ServerName helpdesk.digitalglobe.com
   DocumentRoot /opt/rt3/share/html
   AddDefaultCharset UTF-8

   # this line applies to Apache2+mod_Perl2 only
   PerlModule Apache2 Apache2::compat

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

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

Notice: Request Tracker needs to run on its own virtual host or stand alone host. It is best to set up a second IP Address for the host that Request Tracker will be deployed. In this way, a standard web server and Request Tracker can coexist on the same server. If this method is employed, then there is no need for running Request Tracker on port 81.

Warning: Request Tracker files must be owned by the user name assigned to Apache.
On RedHat Linux this is the **apache** user. Also, All Request Tracker files must also be in the **rt** group. If this is not done, then Request Tracker will not work.

1.4.4. Install mod_perl V2

Request Tracker utilizes the **mod_perl** module to improve performance while executing Perl code. The installation of **mod_perl** is straight forward, as follows:

Download **mod_perl** from http://search.cpan.og/~pgollucci/mod_Perl-2.0.2, and issue the following commands:

```
tar -xzf mod_perl-2.0.2.tar.gz
```

```
cd mod_perl-2.0.2
perl Makefile.PL MP_APXS=/usr/local/apache2/bin/apxs
make && make test
make install
```

Insert: `LoadModule perl_module modules/mod_perl.so` in **/usr/local/apache2/conf/httpd.conf**

1.4.5. Install Request Tracker

By this point, there should now be a working installation of Perl, MySQL and Apache running on the target server. It is critical that everything has been installed and functioning properly, before continuing. If everything is working, then it is now time to install Request Tracker.

IMPORTANT: Upgrading Request Tracker is a very complex process, so please see **Upgrading Request Tracker V2 to Request Tracker V3**, before continuing. Any customizations made after this point would be lost and will have to be reapplied. If this is an upgrade from Request Tracker V2, then follow the instructions in the aforementioned section.

Download Request Tracker from http://www.bestpractical.com, and issue the following commands:

```
tar -xzf rt-3.4.4
cd rt-3.4.4
 ./configure
make testdeps
make fixdeps      (See Note)
make install
```

Note: The `make fixdeps` step may need to be repeated a few times. What is going on here is that this command is downloading Perl modules from CPAN. All told, this command installs over 50 CPAN modules. It should be noted that this is not a truly automated process, and that the installer must respond to questions in the installation process. When installing these modules, answer “yes” when a questioned is asked, and take the default answers on configuring CPAN.

The CPAN installation does not complete for the following modules. The first three modules fail on a couple tests, which should not affect the performance of Request Tracker. They can be installed, as follows:

```
cd /root/.cpan
cd build/libwww-perl-5.803
perl Makefile.PL
make
make install

cd /root/.cpan
cd build/WWW-Mechanize-1.86
perl Makefile.PL
make
make install

cd /root/.cpan
cd build/Test-WWW_Mechanize-1.08
perl Makefile.PL
make
make install
```

The DBD::mysql module does not install, because the local installation of MySQL is installed in **/usr/local/mysql** and not **/usr/local** (hard coded in the Makefile.PL). This module can be installed as follows:

```
perl –MCPAN –e ‘install DBD::mysql’
cd /root/.cpan
cd build/DBD-mysql-3.0002
perl Makefile.PL --mysql_config=/usr/local/mysql/bin/mysql_config
make
make test
make install
```

IMPORTANT: Up to this point the installation of Request Tracker, and prerequisite software, are identical. If this is a new installation, you would need to issue the following command. **DO NOT ISSUE THE FOLLOWNG COMMAND IF YOU PERFORMED AN UPGRADE**:

```
make initialize-database
```

END IMPORTANT

Next, create a new group in **/etc/group** for Request Tracker and call it **rt**. This step is important required by Request Tracker, as is the following command:

```
groupadd rt
chown -R apache:rt /opt/rt3
```

As mentioned in the **Installing Apache** section, if the Request Tracker files are not owned by the web server user, **apache** for Apache V2, and not in the **rt** group, Request Tracker will not function.

Next, customize **/opt/rt3/etc/RT_SiteConfig.pm** and change all the **example.com** entries to the desired domain name. Also, make sure the variable:

Set($RTAddressRegexp , '^rt\@helpdesk.digitalglobe.com$');

has the fully qualified domain name of the host. It is required for e-mail.

The following variables need to be customized for the local site:

```
Set($rtname , ‘digitalglobe.com’);
Set($Organization , ‘digitalglobe.com’);
Set($Timezone , ‘US/Mountain’);
Set($RTAddressRegexp , '^rt\@helpdesk.digitalglobe.com$');
Set($CorrespondAddress , 'correspond@helpdesk.digitalglobe.com');
Set($CommentAddress , 'comment@helpdesk.digitalglobe.com');
Set($WebBaseURL , "http://helpdesk.digitalglobe.com");
Set($DateDayBeforeMonth, 0);
Set($MyTicketLength, 25);
Set($WebDefaultStyleSheet, ‘3.4-compat’);    (RT 3.6.3 Only)
```

Next, set up the mail aliases for Request Tracker in **/etc/aliases**, by inserting the various aliases for the Request Tracker Queues and mail gateway. A sample alias is shown below:

```
rt:         "|/opt/rt3/bin/rt-mailgate --queue general --action \
correspond --url http://helpdesk.digitalglobe.com/"
rt-comment: "|/opt/rt3/bin/rt-mailgate --queue general --action comment \
--url http://helpdesk.digitalglobe.com/”
```

Note: Two aliases are required for each queue set up in Request Tracker. Also, when setting up Queues in Request Tracker, make sure to give all rights to the System Groups: Privileged, Unprivileged and Everyone, or users will not be able to e-mail requests to a particular queue.

Type the following to activate the aliases: `newaliases`

The following is required for Request Tracker e-mail to work:

```
cd /etc/smrsh
ln -s /opt/rt3/bin/rt-mailgate
```

Operational Issue: When installing or modifying source code for Request Tracker, and associated software, the web server needs to be restarted.

IMPORTANT: Note the following two steps!

Install the **Apache::DBI** CPAN which is required by **mod_perl** and Request Tracker (it is not mentioned in the installation):

```
perl –MCPAN –e 'install Apache::DBI'
```

The following link needs to be established, as the PATH is not properly set up by the **mod_perl** installation:

```
cd /usr/local/lib/perl5/site_perl/5.8.7/i686-linux
ln –s Bundle/Apache2.pm
```

END IMPORTANT

Now the Apache web server can be started, as follows:

```
/usr/local/bin/apache2/bin/apachectl start
```

Insert the above command in **/etc/rc.local**, in order to start Apache web server when the system is rebooted.

Before going on, make sure the Apache web server is operational. This can be done simply by typing:

```
ps –ef | grep httpd
```

If there are no **httpd** processes running, then check the Apache error log in **/usr/local/apache2/logs/error_log**. It should be noted, that getting an Apache web server operational with **mod_perl**, is the most challenging part of the installation.

At this point, Request Tracker should be operational. To test, just point a web browser to the URL of the Request Tracker virtual server, e.g. **http://helpdesk.digitalglobe.com**. The Request Tracker Login screen should now be displayed.

1.4.6. Set Up Sendmail – SMTP AUTH Setup on RedHat Enterprise

On RedHat Enterprise Linux systems, sendmail is not enabled for incoming mail. There are detailed instructions, on the set up procedure, in **SMTP AUTH HOW TO** located at:

http://www.simpaticus.com/Linux/sendmail-smtp-auth-howto.php

See the Appendix for an excerpt from the above document describing the procedure.

Note: The sendmail server must be activated so Request Tracker can receive e-mail.

Before setting up SMTP AUTH, the **sendmail-cf** and **sendmail-devel** RPM files need to be installed, to check:

```
rpm –qa | grep sendmail
```

If they are not installed, then issue the following commands:

```
up2date-nox –i sendmail-cf sendmail-devel
```

After **sendmail.cf** is updated, restart sendmail:

```
/etc/init.d/sendmail restart
```

1.4.7. Install RTx::Statistics

RTx::Statistics is an extension to Request Tracker. This extension produces graphs and some statistics regarding Trouble Tickets. The installation process requires several steps, as follows:

1.4.7.1. Prepare the Linux environment

Install required development RPM files, if they have not been installed, as follows:

```
up2date-nox –i freetype-devel
up2date-nox –I libjpeg-devel
```

The **GD Graphics Library** and **libpng** packages, provided with RedHat Enterprise WS 3, are too old to use with CPAN modules required for the Statistics package. These two software packages need to be downloaded and installed.

Download **libpng** from http://www.libpng.org/, and issue the following commands:

```
tar –xzf libpng-1.2.8-config.tar.gz
cd libpng-1.2.8-config
./configure --prefix=/usr/local
make
make install
```

Download **GD Graphics Library** from http://www.boutell.com, and issue the following commands:

```
tar –xzf gd-2.0.33.tar.gz
cd gd-2.0.33.tar.gz
./configure --prefix=/usr/local
make
make install
```

The next requirement is to install the Perl CPAN modules required to communicate between Perl and the GD Graphics Library. This can be accomplished by issuing the following commands:

```
perl –MCPAN -e 'install GD'
perl –MCPAN -e 'install GD::Text'
perl –MCPAN -e 'install GD::Graph'
perl –MCPAN -e 'install HTML::QuickTable'
```

At this point, the Linux environment has been prepared for the installation of Request Tracker Statistics.

1.4.7.2. Installing RTx::Statistics Extension

The next step is to install the RTx::Statistics extension.

Download RTx::Statistics from http://anonftp.mqsoftware.com/kfh/RT/, and issue the following commands:

```
tar-xzf RTx-Statistics-0_1_8.tar.gz
cd RTx-Statistics-0_1_8
perl Makefile.PL
make
make install
```

WARNING: Before performing an upgrade to RTx::Statistics, make sure a copy is made of the files noted below, as there were major revisions from the original software package.

1.4.7.3. Patches for RTx::Statistics V0.1.8

In addition to modifying **index.html** (see section on **Spreadsheet Functionality**) to support spreadsheets for OpenOffice, there were several other changes required in RTX::Statistics. These changes fix a couple logic errors in the software. Essentially, the patches provide changes to the logic that will allow totals to display properly. The author did not provide the correct logic for Request Tracker V3.

/opt/rt3/local/lib/RTx/Statistics.pm

The graphs are set to only display 10 days of information and are rather small. The following changes will allow the graphs to display more data in a larger format.

Change the following:

`$GraphWidth` from 500 to 700
`$GraphHeight` from 400 to 600
`$MultiQueueMaxRows` from 10 to 31
`$PerDayMaxRows` from 10 to 31
`$TimeToResolveMaxRows` from 10 to 31
`$MultiQueueLabelDateFormat` from %a to %F
`$PerDayLabelDateFormat` from %a to %F
`$TimeToResolveLabelDateFormat` from %a to %F

The following scripts had major changes:

/opt/rt3/share/html/RTx/Statistics/CallsMultiQueues/Elements/Chart
/opt/rt3/share/html/RTx/Statistics/CallsQueueDay/Results.tsv
/opt/rt3/share/html/RTx/Statistics/CallsQueueDay/index.html
/opt/rt3/share/html/RTx/Statistics/CallsQueueDay/Elements/Chart
/opt/rt3/share/html/RTx/Statistics/index.html
/opt/rt3/share/html/RTx/Statistics/Elements/Tabs
/opt/rt3/share/html/RTx/Statistics/Elements/StatColumnMap

WARNING: Before installing a new version of **RTx::Statistics**, make a copy of the above files.

Once RTx::Statistics is installed and the patches applied, then do the following:

```
chown -R apache:rt /opt/rt3/*
/usr/local/apache2/bin/apachectl stop
rm -rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

On the next login to Request Tracker, a new menu item called **RTx::Statistics** will be displayed. When selecting various items from the **RTx::Statistics** area, a graph should be displayed with most of the statistic report pages.

1.4.8. Install RTx::RightsMatrix

RTx::RightMatrix enables the user to main the various Request Tracker and Asset Tracker privileges and rights from a single management area. This package utilizes a simple matrix in order to maintain the rights and privileges. Once installed, it can be access via:

Configuration -> Tools -> Rights Matrix

The installation of RTx::RightsMatrix is very simple, by doing the following:

```
perl -MCPAN -e ‘install RTX::RightsMatrix
```

Next, issue the following commands:

```
chown -R apache:rt /opt/rt3/*
/usr/local/apache2/bin/apachectl stop
rm -rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

On the next login to Request Tracker, **RTx::RightMatrix** will be available.

1.4.9. Install Request Tracker FAQ Manual (RTFM)

The Request Tracker FAQ Manual or RTFM, is a tool that allows the capability of creating a Knowledgebase within the Request Tracker environment. RTFM is very easy to install and maintain. Also, Custom Fields can be created to support the RTFM environment.

To install, download RTFM from http://www.bestpractical.com, and issue the following commands:

```
Perl -MCPAN -e ‘install YAML’

tar -xzf RTFM-2.1.40.tar.gz
cd RTFM-2.1.40
perl Makefile.PL
make initdb
make install

/usr/local/apache2/bin/apachectl stop
rm -rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

Note: Customization and usage information is in the *Request Tracker FAQ Manual System Manager’s User’s Guide.*

1.4.10. Install RTx::AssetTracker

The final component to install is RTx::AssetTracker, which is a powerful Inventory Management extension to Request Tracker.

Download RTx::AssetTracker from http://atwiki.chaka.net/, and issue the following commands:

```
tar -xzf at-1.2.2.tar.gz
cd at-1.2.2
./configure --with-rt-home=/opt/rt3 --with-rt-local=/opt/rt3 \
--with-rt-localhtml=/opt/rt3/share/html --with-db-admin=root
make
make install
/usr/local/apache2/bin/apachectl stop
rm –rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

Installation Note: During the `make install` process, the installation will ask for the MySQL root password, enter return is none is set or enter the proper password.

At this point the RTx; AssetTracker module is installed, including the schema for the RTx: AssetTracker database.

Operational Issue: When installing or modifying source code for Request Tracker, and associated software, the web server needs to be restarted.

1.4.10.1. Set Up Ticket to Asset Link Scrip

A freeware Scrip has been created to allow for the easy linking of a Trouble Ticket to a particular asset. The procedure is as follows:

Download **ExtractCustomFieldValues** from http://wiki.bestpractical.com/index.cgi?Contributions

Next, issue the following commands:

```
tar –xzf ExtractCustomFieldValues-1.0b1.tgz
cd ExtractCustomFieldValues-1.0b1
make install
```

Note: Check the `Makefile` to be sure that the PATHs are set correctly.

Once, **ExtractCustomFieldValues** is installed, the next thing to do is install the **LinkTicketToAsset** Scrip, which is located at:

http://atwiki.chaka.net/index.cgi?HowToLinkTicketToAssetScrip

A copy of the Scrip is also included in the Appendix.

Operational Issue: When installing or modifying source code for Request Tracker, and associated software, the web server needs to be restarted.

To install first log into Request Tracker, and do the following:

Select **Configuration** -> **Global** -> **Scrips -> New Scrip**

Follow the instructions on the above web page or in the Appendix; it is fairly straight forward on where to put the information on the page. Also, change **server_name** to **Asset** in the **Custom condition** block, and save the Scrip. The reason for the change is for consistency purposes.

Next, select **Configuration** -> **Custom Fields** -> **New Custom Fields**

Create an **Asset** custom field and place it in the **Ticket** category (on pull down menu).

Next, select **Configuration** -> **Custom Fields** -> **Asset**

Under **Applies to,** select all the Queues.

Under **Group Rights** and **User Rights**, select all of the privileges, this will activate the new **Asset** Trouble Ticket field.

Once this is implemented, when a trouble ticket is displayed, a new Custom Field will appear just below **The Basics** box on the trouble ticket. When working on a trouble ticket to assign the asset, use **Jumbo** mode and enter the desired asset name. When the trouble ticket is saved, a bi-directional link will be created between the trouble ticket and the asset.

One note, if new queues are added to Request Tracker, then do the following:

Next, select **Configuration** -> **Custom Fields** -> **Asset**
Under **Applies to,** select all the Queues.

This procedure must be executed to add the **Asset** field to Trouble Tickets destined to the new queue.

1.4.11. Possible Installation Issue

If this were an upgrade, then the Request Tracker database should have been migrated and updated by this point. If not, then installing the following modules may have unexpected results. RTX::AssetTracker and **ExtractCustomFieldValues** make modifications to the Request Tracker database to accommodate the new features. Therefore, perform the database upgrade and data migration, before installing the following modules.

1.4.12. Additional Installation Notes

Upon completion of the above installation, the Request Tracker software is ready to configure. The default login is **root** for the user name and **password** for the password. The first thing that should be done is change the password to something more secure.

Once the **root** account is secure, then users, asset types, queues, and groups need to be created. In addition, for every new queue created, an entry in `/etc/aliases` is required for e-mail purposes.
If changes are made to the Request Tracker software and/or extensions, then the web server needs to be restarted. This is required because mod_perl caches the Request Tracker code, to improve performance. It is for this reason that Request Tracker should be installed on its own dedicated server.

2. Upgrading RT V2 to RT V3

2.1. Introduction

The upgrade from Request Tracker V2 to Request Tracker V3 is a complex process. This section supplements the **Installation of Request Tracker** document. The installer is advised to read through both documents, before attempting an installation or upgrade of Request Tracker. If the steps, outlined in these documents, are not properly followed, then the upgrade of Request Tracker V2 will fail.

This document assumes that Request Tracker V3 is being deployed on another host than the location of Request Tracker V2. This is advised, in the event one has problems with the Request Tracker V3 upgrade.

2.2. Preparing for Migration

2.2.1. Install Request Tracker V2

Request Tracker V2 needs to be installed for the migration of Request Tracker V2 formatted data to Request Tracker V3 formatted data. Request Tracker V2s libraries are used as part of the data migration process. Once the migration is complete, the software can be removed.

In order to install Request Tracker V2, mod_perl V1 also needs to be installed. Download and install mod_perl V1 from http://search.cpan.org/~gozer/mod_erpl-1.29/ and install the software:

```
tar -xzf mod_perl-1.29.tar.gz
cd mod_perl-1.29
perl Makefile.PL
make && make test
make install
```

Download and install Request Tracker V2 from http://www.bestpractical.com and install the software

```
tar -xzf rt-2-0-15.tar.gz
cd rt-2-0-15
./onfigure
make
make testdeps
make fixdeps
```

The installation of libapreq will fail even though mod_perl v1 was installed (libareq is required for Apache::Cookie). There is an error in the `Makefile.PL` that needs to be corrected, as follows:

```
cd /root/.cpan/build/libapreq-1.33
```

Comment out the lines that check for mod_perl V1 (lines 26 - 36). The whole BEGIN block encompassing the code:

```
#BEGIN {
#    unless (eval {require mod_perl}) {
#        die "Please install mod_perl: 1.25 < version < 1.99\n($@)";
#    }
#    if ($mod_perl::VERSION < 1.2402) {
```

```
#        die "Please upgrade mod_perl to 1.24_02 or greater";
#    }
#    elsif ($mod_perl::VERSION > 1.98) {
#        die "mod_perl 1.x ( < 1.99) is required";
#    }
#}
```

Then, issue the following commands:

```
perl Makefile.PL
make install
```

Once libapreq is installed, then issue the following commands:

```
make fixdeps
make install
```

An instance of Request Tracker 2 is now installed.

2.2.2. Importing the Request Tracker V2 Database

There are several ways to import a MySQL Database. This method is the most easiest to use. However, the database server should be shut down on the server hosting the Request Tracker V2 Database and the new server for the Request Tracker V3 database. So, shutdown the database server as follows:

```
/etc/init.d/mysqld stop
```

Now, copy the database from the odl server to the new server:

```
rm /usr/local/mysql/var/rt2/*
cd /usr/local/mysql/var/rt2
scp old-database-host:/usr/local/mysql/var/rt2/* .
chown mysql:mysql *
chmod 664 *
```

Restart the database server:

```
/etc/init.d/mysqld start
```

Special Note: Sometimes a database table may get corrupted, due to improper maintenance. The following command is an example of performing a repair on all of the Request Tracker V2 database tables:

```
/usr/local/mysql/bin/mysql –u root –p
mysql> use rt2;
mysql> repair table rt2.ACL;
mysql> repair table rt2.Attachments;
mysql> repair table rt2.GroupMembers;
mysql> repair table rt2.Groups;
mysql> repair table rt2.Links;
mysql> repair table rt2.Queues;
mysql> repair table rt2.ScripActions;
mysql> repair table rt2.ScripConditions;
mysql> repair table rt2.Scrips;
mysql> repair table rt2.Templates;
mysql> repair table rt2.Tickets;
mysql> repair table rt2.Transactions;
mysql> repair table rt2.Users;
mysql> repair table rt2.Watchers;
```

2.2.3. Request Tracker and Database Migration

The upgrade path to Request Tracker V3.4.4 is a multi-step process and requires the installation of Request Tracker V3.0.2, Request Tracker V3.2.3 and Request Tracker V3.4.4. Each of these installations, require the installer to install a Perl CPAN module, install the requisite Request Tracker software version and update the database schema.

In the event, that Request Tracker V3.4.4 was already installed, then save the copy of /opt/rt3. Chances are that Request Tracker V3.4.4 was already installed and is working, as per the **Installation of Request Tracker** documentation.

```
mv /opt/rt3 /opt/rt3.new
```

In order for the migration to succeed, three different versions of **DBIx-SearchBuilder** must be installed with each installation of Request Tracker V3.

Download DBIx-SearchBuilder-1.01.tar.gz (for rt 3.0.12), DBIx-SearchBuilder-1.22.tar.gz (for rt 3.2.3) and DBIx-SearchBuilder-1.35.tar.gz (for rt-3.4.4) from http://search.cpan.org/~jesse. V 1.01 is required for the data Request Tracker V2 to Request Tracker V3 export/import, V 1.22 is required for the upgrade from Request Tracker V3.0.12 to Request Tracker V3.2.3, and V1.35 is required for the upgrade from Request Tracker V3.2.3 to Request Tracker V3.4.4.

2.2.4. First Stage Database Migration

2.2.4.1. Install Request Tracker V3.0.12

```
tar -xzf DBIx-SearchBuilder-1.01.tar.gz
cd DBIx-SearchBuilder-1.01
perl Makefile.PL
make
make install
```

Now, download Request Tracker V3.0.12 from http://www.bestpractical.com and install it.

```
tar -xzf rt-3.0.12.tar.gz
cd rt-3.0.12
./configure --with-web-user=apache --with-web-group=apache
make
make install
make initialize-database
```

2.2.4.2. Request Tracker V2 to Request Tracker V3.0.12 Database Migration

At this point, the Request Tracker V2 database can be migrated to the first Request Tracker V3 release. Best Practical provided a database migration tool, which will upgrade a Request Tracker V2 database to Request Tracker V3 format.

Download the database migration tool (rt2-to-rt3.tar.gz) from http://www.bestparctical.com and install the software:

```
tar -xzf rt2-to-rt3.tar.gz
cd rt2-to-rt3
```

Modify rt-2.0-to-dumpfile to have `libs` point to `/opt/rt2`

```
use lib "/opt/rt2/lib";
use lib "/opt/rt2/etc";
```

Modify dumpfile-to-rt-3.0 to have `libs` point to `/opt/rt3`

```
use lib "/opt/rt3/lib";
use lib "/opt/rt3/etc";
```

Create a directory to store the Request Tracker V2 database dump:

```
mkdir rt2_data
```

Perform the database dump:

```
./rt-2.0-to-dumpfile rt2_data
```

FYI: This takes about 90 minutes for approximately 15000 tickets.

Perform the data migration:

```
./dumpfile-to-rt-3.0 rt2_data
```

FYI: This also takes 90 minutes for approximately 15000 tickets.

Once the database is migrated, next Request Tracker V3.0.12 needs to be upgraded to Request Tracker V3.2.3

2.2.5. Second Stage Database Migration

2.2.5.1. Install Request Tracker V3.2.3

First upgrade DBIx-SearchBuilder:

```
tar -xzf DBIx-SearchBuilder-1.22.tar.gz
cd DBIx-SearchBuilder-1.22
perl Makefile.PL
make
make install
```

Now, download rt-3.2.3.tar.gz from http://www.bestpractical.com and install it.

```
tar -xzf rt-3.2.3.tar.gz
cd rt-3.2.3
./configure --with-web-user=apache --with-web-group=apache
make
make install
make upgrade
```

Now perform the database upgrade, as follows:

```
cd rt-3.2.3/etc/upgrade
/opt/rt3/bin/rt-setup-database -dba root -prompt-for-dba-password \
-action schema -datadir 3.1.0
/opt/rt3/bin/rt-setup-database -dba root -prompt-for-dba-password \
-action acl -datadir 3.1.0
```

```
/opt/rt3/bin/rt-setup-database -dba root -prompt-for-dba-password \
-action insert -datadir 3.1.0
```

If this works, then Request Tracker V3.2.3 needs to be upgraded to Request Tracker V3.4.4

2.2.6. Third Stage Database Migration

2.2.6.1. Install Request Tracker V3.4.4

First upgrade DBIx-SearchBuilder:

```
tar -xzf DBIx-SearchBuilder-1.35.tar.gz
cd DBIx-SearchBuilder-1.35
perl Makefile.PL
make
make install
```

Now, download rt-3.4.4.tar.gz from http://www.bestpractical.com and install it.

```
tar -xzf rt-3.4.4.tar.gz
cd rt-3.4.4
./configure --with-web-user=apache --with-web-group-apache
make
make install
make upgrade
```

Now perform the database upgrade, as follows:

```
cd rt-3.4.4/etc/upgrade
/opt/rt3/bin/rt-setup-database -dba root -prompt-for-dba-password \
-action schema -datadir 3.3.0
/opt/rt3/bin/rt-setup-database -dba root -prompt-for-dba-password \
-action acl -datadir 3.3.0
/opt/rt3/bin/rt-setup-database -dba root -prompt-for-dba-password \
-action insert -datadir 3.3.0
```

At this point, Request Tracker has been upgraded to Request Tracker V3.4.4.

2.3. Post Upgrade

2.3.1. Root Account Privileges

The Request Tracker upgrade does not transfer privileges properly from Request Tracker V2 and through the subsequent upgrades. Also, new privileges have been added, so there are additional privileges required to have full configuration capability over Request Tracker.

A shell script has been created to give **root** full privileges, in order to work with Request Tracker V3. Also, the script will reset the root password to **password**, for ease of login. The root password should be changed accordingly upon the first login to Request Tracker V3.

3. Upgrading Request Tracker to V3.6.3

3.1. Introduction

Unlike the upgrade process of Request Tracker from V2 to V3, the upgrade from Request Tracker V3.4.4 to Request Tracker V3.6.3 is a very simple and straight forward process. As there were few modifications to the Request Tracker V3.4.4 source code, the installer should be able to have Request Tracker V3.6.3 available for production in a relatively short time period. This section is designed to provide the installer with the information to perform a successful upgrade.

3.2. Install Request Tracker 3.6.3

WARNING: Before proceeding, make sure that a backup of the database has been performed. Also, make sure to save a copy of **/opt/rt3/etc/RT-Config.pm** and **/opt/rt3/etc/RT_SiteConfig.pm**. Also, make sure that both the web server and sendmail have been shut down properly.

Shutdown down the web server and sendmail:

```
/usr/local/apache2/bin/apachectl stop
/etc/init.d/sendmail stop
```

The command to backup the database is:

```
/usr/local/mysql/bin/mysqldump --databases rt3 > /tmp/rt3.sql
```

In the event of a database corruption, to restore a database do the following:

```
/usr/local/mysql/bin/mysql –u root
mysql> drop database rt3;
mysql> exit
/usr/local/mysql/bin/mysql –u root < /tmp/rt3.sql
```

As described in the previous section, a new release of Request Tracker needs to be downloaded from the Best Practical web site. So, the installer needs to download **rt-3.6.3.tar.gz** from http://www.bestpractical.com and do the following:

```
tar -xzf rt-3.6.3.tar.gz
cd rt-3.6.3
./configure --with-web-user=apache --with-web-group-apache
make
make testdeps
make fixdeps
make upgrade
```

Now perform the database upgrade, as follows:

```
cd rt-3.6.3/etc/upgrade/3.5.1
```

Next, make the following change to **content**:

In both occurrences of the line beginning with `Query`, add “`OR Status = ‘stalled’` after `Status = ‘open’`.

Also, remove the following line:

```
{ type => 'component', name => 'QuickCreate' },
```

As the **QuickCreate** feature is very confusing to users, by removing this line, the **QuickCreate** feature is not set up universally. If a user desires the **QuickCreate** feature, they can easily add it by using the edit feature on the “RT at a glance” page.

Now issue the following commands

```
cd ..
/opt/rt3/bin/rt-setup-database –dba root –prompt-for-dba-password \
–action insert –datadir 3.5.1
```

At this point, Request Tracker has been upgraded to Request Tracker V3.6.3.

3.3. Post Upgrade

Once Request Tracker V3.6.3 is installed and the database has been upgraded, then the installer just needs to perform some of the customizations outlined in the next chapter. The sections applicable to the V3.6.3 are as follows:

1. **4.1 Spreadsheet Functionality**
2. **4.6 Adding a notification concerning laptop computers**
3. **4.7 Quick Links to Request Tracker Queues**
4. **4.8 Additional “Save Changes” Capability**
5. **4.9 Making Time Worked a mandatory field upon Resolve**
6. **4.10 Restrict “newest unowned tickets” display**
7. **4.11 Drop Down List Window Size Increase**
8. **4.12 The guest Account User menu**
9. **4.13 Adding “stalled” tickets to various ticket displays**

The following section is optional:

1. **4.4 Turning off automatic display of images in Ticket Display** – This feature is no longer warranted since the new anti-SPAM procedures have been put in affect at DigitalGlobe.

The following sections are no longer applicable:

1. **4.2 Stalled Ticket on Quicksearch** – This item has been restored to Request Tracker V3.6.3.
2. **4.3 Clickable Hypertext Links in Tickets** – This feature is not part of Request tracker V3.6.3.
3. **4.5 Displaying Stalled Tickets in “25 highest priority”** – This feature is user customizable in Request Tracker V3.6.3.

In regards to RTx::AssetTracker, RTx::Statistics and RTx::RightsMatrix, these applications require no change from the Request Tracker V3.4.4 installation. As these extensions are already in the Request Tracker directory structure, they are unaffected by the Request Tracker upgrade.

A Special Note: With the introduction of Request Tracker V3.6.3, the authors introduced a new display interface. This new interface is more modern and less formal looking than the interface provided with Request Tracker V3.4.x. Request Tracker V3.6.3 does provide emulation mode for the Request Tracker V3.4.x interface, by adding the following line to **/opt/rt3/etc/RT_Siteconfig.pm**:

```
Set($WebDefaultStylesheet, ‘3.4-compat’ );
```

Once all the customizations have been made, the following commands need to be executed:

```
/usr/local/apache2/bin/apachectl stop
rm –rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

If sendmail was shutdown, then issue the following command:

```
/etc/init.d/sendmail start
```

At this point, the installer should be able to log into Request Tracker and start working with Request Tracker V3.6.3.

Notice: The installer should review **/opt/rt3/etc/RT_Config.pm** and **/opt/rt3/etc/RT_SiteConfig.pm** to make sure that nothing changed in the configuration. It is always advisable to save these files, before attempting an upgrade. Also, local configuration changes should only be made to **/opt/rt3/etc/RT_Siteconfig.pm**.

4. Customizations

4.1. Spreadsheet Functionality

When running Request Tracker on a Windows system, one can get a seamless interface to Excel. That is, when one “clicks” on the **spreadsheet** option, Request Tracker calls Excel, open up Excel and sends the desired data to Excel. If the same thing is done on a Linux or UNIX system, using OpenOffice, than the data is sent to OpenOffice Writer and not OpenOffice Calc.

What is happening is Request Tracker is generating an output file with a **tsv** extension, as the data is produced in a Tab Separated Value format. Microsoft Excel recognizes a **tsv** file with no problem; however, OpenOffice Calc does not. So, OpenOffice converts the results to an ordinary text file, as it does not “support” the **tsv** file extension. However, OpenOffice does support the **csv** (Comma Separated Values) extension. There is a work around for this, as follows:

1. When one “clicks” on **spreadsheet**, than save the file to disk; it will be called **Results.tsv**, **Results-1.tsv**, etc.
2. Rename **Results.tsv** to **Results.csv**.
3. Run OpenOffice Calc and open **Results.csv**. Calc will ask for format of the file, in the **Separated By** section of the pop-up screen, click on **Tab** and uncheck **Comma**.
4. Once these steps are followed, Calc will display the desired data in spreadsheet format.

The following steps can be taken, to create a seamless interface for UNIX and Linux users:

```
cd /opt/rt3/share/html/Search
ln –s Results.tsv Results.csv
```

Edit `index.html` and make the following changes:

```
<a href="<%$RT::WebPath%>/Search/Results.tsv<%$QueryString%>"> \
<&|/l&>spreadsheet (Excel)</&></a> |
<a href="<%$RT::WebPath%>/Search/Results.csv<%$QueryString%>"> \
<&|/l&>spreadsheet (OpenOffice)</&></a> |
```

Another change, for Results.html, changes the order of the Ticket display on the results page. Change line 138 to read: `$Order = ‘DESC’`

Notice: The user should be made aware that **spreadsheet (Excel)** will only work on Windows systems, and **spreadsheet (OpenOffice)** will only work on Linux and UNIX systems.

If AssetTracker is also installed, then do the following:

```
cd /opt/rt3/share/html/AssetTracker/Search
ln –s Results.tsv Results.csv
```

Edit `index.html` and make the following changes:

```
<a href="<%$RT::WebPath%>/AssetTracker/Search/Results.tsv<%$QueryString%>"> \
<&|/l&>spreadsheet (Excel)</&></a><br>
<a href="<%$RT::WebPath%>/AssetTracker/Search/Results.csv<%$QueryString%>"> \
<&|/l&>spreadsheet (OpenOffice)</&></a><br>
```

Issue the following commands:

```
/usr/local/apache2/bin/apachectl stop
rm –rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

If RTx::Statistics is also installed, then do the following:

```
cd /opt/rt3/share/html/RTx/Statistics/CallsQueueDay
ln –s Results.tsv Results.csv
```

Edit `index.html` and make the following changes:

```
<a href="<%$RT::WebPath%>/RTx/Statistics/CallsQueueDay/Results.tsv<%$QueryString%>"> \
<&|/l&>spreadsheet (Excel)</&></a><br>
<a href="<%$RT::WebPath%>/RTx/Statistics/CallsQueueDay/Results.csv<%$QueryString%>"> \
<&|/l&>spreadsheet (OpenOffice)</&></a><br>
```

```
cd /opt/rt3/share/html/RTx/Statistics/OpenStalled
ln –s Results.tsv Results.csv
```

Edit `index.html` and make the following changes:

```
<a href="<%$RT::WebPath%>/RTx/Statistics/SpenStalled/Results.tsv<%$QueryString%>"> \ <&|/
l&>spreadsheet (Excel)</&></a><br>
<a href="<%$RT::WebPath%>/RTx/Statistics/OpenStalled/Results.csv<%$QueryString%>"> \
<&|/l&>spreadsheet (OpenOffice)</&></a><br>
```

Issue the following commands:

```
/usr/local/apache2/bin/apachectl stop
rm –rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

Notice: The user should be made aware that **spreadsheet (Excel)** will only work on Windows systems, and **spreadsheet (OpenOffice)** will only work on Linux and UNIX systems.

4.2. Stalled Ticket Display on Quicksearch

The default Request Tracker V3 installation only displays New and Open Tickets, on the Quick search screen (the RT at a glance display). Apply the following patch **to /opt/rt3/share/html/Element/Quicksearch**:

```
46,52d45
< %#RAL BEGIN RAL TAGGED BLOCK }}}
< %#RAL Modification:
< %#RAL 05-DEC-2005 - Nick Metrowsky
< %#RAL         Patch courtesy of Stephen Dowdy, NCARD
< %#RAL         Display a lsit of Stalled Tickets
< %#RAL         on the Request Tracker Quicksearch page.
< %#RAL END RAL TAGGED BLOCK }}}
60d52
<        <th class="collection-as-table" align=right><&|/l&>Stall</&></th>
73,74c65
<   my $stall_q = "Queue = '".$queue->Name."' AND Status = 'stalled'";
<   my $all_q = "Queue = '".$queue->Name."' AND (Status = 'open' OR Status = 'new' OR
Status = 'stalled')";
---
>   my $all_q = "Queue = '$name' AND (Status = 'open' OR Status = 'new')";
82,84d72
```

```
<   $Tickets->FromSQL($stall_q);
<   my $stall = $Tickets->Count();
<
91d78
< <td align="right"><A HREF="<% $RT::WebPath%>/Search/Results.html?Query=<%$stall_q |nu
%>&Rows=50"><%$stall%></a></TD>
```

Next, issue the following commands:

```
/usr/local/apache2/bin/apachectl stop
rm -rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

On the next login to Request Tracker, a **Stall** column will now be displayed.

4.3. Clickable Hypertext Links in Tickets

Request Tracker does not allow a hypertext link within the body of Ticket. There are some situations which having an active hypertext link would make it easier for technicians to troubleshoot World Wide Web problems. Fortunately, a small patch is available at http://wiki.bestpractical.com, which adds this feature. This patch is actually defined as a Callback.

To obtain the patch, go to http://wiki.bestpractical.com and search for **ClickableLinks**.

Click on **ClikableLinks** and insert the following code into a new file, as follows:

```
/opt/rt3/local/html/Callbacks/MI/Ticket/Elements/ShowMessageStanza/Default
```

Contents of `Default`:

```
<%init>
my $val = $$content;
use bytes;
$val =~ s/https?:([\w\/\@\-~%#?=;,.:+]|\&)+[\w\/]/<a href="$&" target="_blank">$&<\/a>/gi;
$val =~ s/(s?ftp|ftps):[\w\/\.:+\-]+/<a href="$&">$&<\/a>/gi;
$val =~ s/(s?file):[\w\/\.:+\-]+/<a href="$&">$&<\/a>/gi;
$val =~ s/(mailto:)?([a-zA-Z0-9_\+\-\.]+\@([a-zA-Z0-9][\w\.\+\-]+\.[a-zA-Z]{2,})(\?
([\w\/\@\-~%#?=;,.:+]|\&)+[\w\/])?)/<a href="mailto:$2">$&<\/a>/gi;
$val =~ s/\<((url:\s*)?\w+:[^&<>]+)\>/<<a href="$1" target="_blank">$1<\/a>>/gi;
$$content = $val;
</%init>
<%args>
$content => undef
</%args>
```

Note: The second “%val” is line wrapped, and it and the line beginning with “z” should be on one line.

After installing the above code, issue the following commands:

```
/usr/local/apache2/bin/apachectl stop
rm -rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

4.4. Turning off automatic display of images in Ticket Display

Request Tracker provides the capability of displaying an image that was submitted as part of the ticket. Normally, this feature is used to send screen shots to provide additional information to the technician processing the ticket. However, because SPAM e-mail messages can also be sent to Request Tracker, this feature allows for the display of some unpleasant images. The following patch will disable the automatic image display feature, but still allows the technician to display the image by clacking on a link within a ticket.

To turn off the automatic display of images within a ticket, modify the following script:

/opt/rt3/share/html/Ticket/Elements/ShowTransactionAttachments :

```
#    elsif ( $message->ContentType =~ /^image\//i ) {
#        $m->out('<img src="'
#              . $AttachPath . '/'
#              . $Transaction->Id . '/'
#              . $message->Id
#              . '/">' );
#    }
```

After installing the above code, issue the following commands:

```
/usr/local/apache2/bin/apachectl stop
rm –rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

4.5. Displaying Stalled Tickets in "25 highest priority…"

Request Tracker only displays **New** and **Open** Tickets in the section "**25 highest priority tickets I own..**" on the main Request Tracker home page. Fortunately, it is a simple change to also include **Stalled** Tickets. The following file needs to be updated:

/opt/rt3/share/html/Elements/MyTickets

```
62c62
> my $Query = " Owner = '".session{'CurrentUser'}->Id."' AND (Status = 'new' OR Status =
'open');
---
< my $Query = " Owner = '".session{'CurrentUser'}->Id."' AND (Status = 'new' OR Status =
'open' OR Status = 'stalled');
```

4.6. *Adding a notification concerning laptop computers*

An issue has arisen with laptop computers, wireless connectivity and undocking/moving laptop computers to another location. Under certain circumstances, the connectivity to the Request Tracker database can be lost due to a possible corruption of a Request Tracker cookie session identifier and MySQL database handle. The solution is for users to logout of Request Tracker before undocking and moving a laptop to a new location.

The following modification was made to **/opt/rt3/share/html/index.html** to warn laptop users about this above issue:

```
95a96,102 (For RT 3.6.3: 114a115,123)
> <HR>
> <b>NOTICE:</b> If you are using any portion of the Request Tracker Suite from a
> laptop PC, and you plan to undock and/or move the laptop to a different location, please
> logoff from Request Tracker BEFORE relocating the laptop. Under certain
> circumstances, Request Tracker may not function properly as network connectivity may
> be lost during the undock/move and this could cause a database connectivity error.
> <HR>
```

The following modification was made to the Guest User Account Menu (**/opt/rt3/share/html/SelfService/CreateTicketInQueue.html**) to warn laptop users about this above issue:

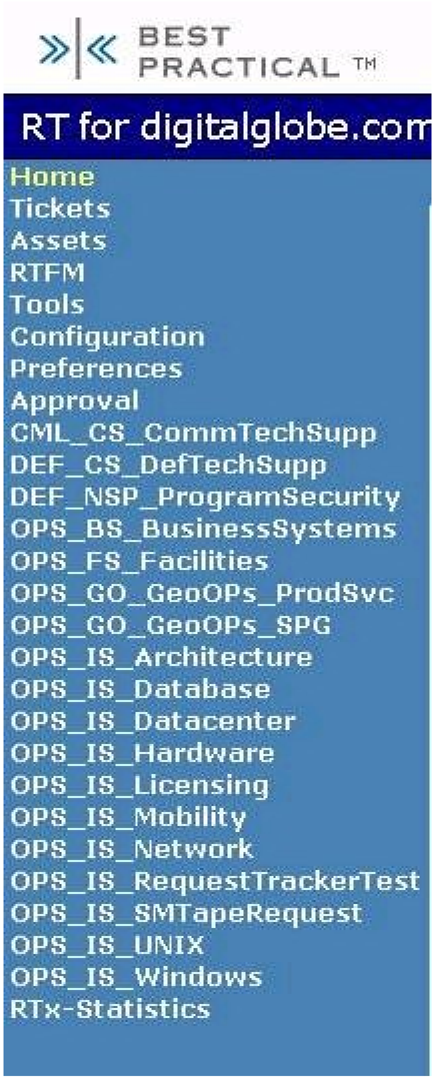
```
57a110,116
> <HR>
> <b>NOTICE:</b> If you are using any portion of the Request Tracker Suite from a
> laptop PC, and you plan to undock and/or move the laptop to a different location, please
> logoff from Request Tracker BEFORE relocating the laptop. Under certain
> circumstances, Request Tracker may not function properly as network connectivity may
> be lost during the undock/move and this could cause a database connectivity error.
> <HR>
```

NOTICE: Please see the next section for information concerning modifications to the Guest User Account Menu.

4.7. Quick Links to Request Tracker Queues

Request Tracker navigation requires that a user must be at the main Request Tracker screen to navigate between queues. It has been determined that this is very inefficient and disruptive to a user's work flow. Therefore, a simple change was made to the Request Tracker menu and style sheet to enable the capability to switch between Request Tracker queues, without the need of returning to the main Request Tracker menu.

The results of the change are depicted in the following image:



The changes were made to the following files:

/opt/rt3/share/html/Elements/Tabs (Request Tracker V3.4.4 only)

```
76a77,131    (For Request Tracker V3.6.3: 79a80,134)
>                     ZA => { title => loc('CML_CS_CommTechSupp'),
>                            path => 'Search/Results.html?
Query=Queue=\'CML_CS_CommTechSupp\' AND (Status = \'open\' OR Status = \'new\' OR Status
= \'stalled\')&Rows=50'
>                          },
>                     ZB => { title => loc('DEF_CS_DefTechSupp'),
>                            path => 'Search/Results.html?
Query=Queue=\'DEF_CS_DefTechSupp\' AND (Status = \'open\' OR Status = \'new\' OR Status =
\'stalled\')&Rows=50'
>                          },
>                     ZC => { title => loc('DEF_NSP_ProgramSecurity'),
>                            path => 'Search/Results.html?
Query=Queue=\'DEF_NSP_ProgramSecurity\' AND (Status = \'open\' OR Status = \'new\' OR
Status = \'stalled\')&Rows=50'
>                          },
>                     ZD => { title => loc('OPS_BS_BusinessSystems'),
>                            path => 'Search/Results.html?
Query=Queue=\'OPS_BS_BusinessSystems\' AND (Status = \'open\' OR Status = \'new\' OR
Status = \'stalled\')&Rows=50'
>                          },
>                     ZE => { title => loc('OPS_FS_Facilities'),
>                            path => 'Search/Results.html?
Query=Queue=\'OPS_FS_Facilities\' AND (Status = \'open\' OR Status = \'new\' OR Status
= \'stalled\')&Rows=50'
>                          },
>                     ZF => { title => loc('OPS_GO_GeoOPs_ProdSvc'),
>                            path => 'Search/Results.html?
Query=Queue=\'OPS_GO_GeoOps_ProdSvc\' AND (Status = \'open\' OR Status = \'new\' OR
Status = \'stalled\')&Rows=50'
>                          },
>                     ZG => { title => loc('OPS_GO_GeoOPs_SPG'),
>                            path => 'Search/Results.html?
Query=Queue=\'OPS_GO_GeoOps_SPG\' AND (Status = \'open\' OR Status = \'new\' OR Status
= \'stalled\')&Rows=50'
>                          },
>                     ZH => { title => loc('OPS_IS_Architecture'),
>                            path => 'Search/Results.html?
Query=Queue=\'OPS_IS_Architecture\' AND (Status = \'open\' OR Status = \'new\' OR Status
= \'stalled\')&Rows=50'
>                          },
>                     ZI => { title => loc('OPS_IS_Database'),
>                            path => 'Search/Results.html?Query=Queue=\'OPS_IS_Database\'
AND (Status = \'open\' OR Status = \'new\' OR Status = \'stalled\')&Rows=50'
>                          },
>                     ZJ => { title => loc('OPS_IS_Datacenter'),
>                            path => 'Search/Results.html?
Query=Queue=\'OPS_IS_Datacenter\' AND (Status = \'open\' OR Status = \'new\' OR Status
= \'stalled\')&Rows=50'
>                          },
>                     ZK => { title => loc('OPS_IS_Hardware'),
>                            path => 'Search/Results.html?Query=Queue=\'OPS_IS_Hardware\'
AND (Status = \'open\' OR Status = \'new\' OR Status = \'stalled\')&Rows=50'
>                          },
>                     ZL => { title => loc('OPS_IS_Licensing'),
>                            path => 'Search/Results.html?
Query=Queue=\'OPS_IS_Licensing\' AND (Status = \'open\' OR Status = \'new\' OR Status
= \'stalled\')&Rows=50'
>                          },
>                     ZM => { title => loc('OPS_IS_Mobility'),
>                            path => 'Search/Results.html?Query=Queue=\'OPS_IS_Mobility\'
AND (Status = \'open\' OR Status = \'new\' OR Status = \'stalled\')&Rows=50'
>                          },
>                     ZN => { title => loc('OPS_IS_Network'),
>                            path => 'Search/Results.html?Query=Queue=\'OPS_IS_Network\'
AND (Status = \'open\' OR Status = \'new\' OR Status = \'stalled\')&Rows=50'
>                          },
>                     ZO => { title => loc('OPS_IS_RequestTrackerTest'),
>                            path => 'Search/Results.html?
Query=Queue=\'OPS_IS_RequestTrackerTest\' AND (Status = \'open\' OR Status = \'new\' OR
Status = \'stalled\')&Rows=50'
>                          },
>                     ZP => { title => loc('OPS_IS_SMTapeRequest'),
>                            path => 'Search/Results.html?
Query=Queue=\'OPS_IS_SMTapeRequest\' AND (Status = \'open\' OR Status = \'new\' OR Status
= \'stalled\')&Rows=50'
>                          },
>                     ZQ => { title => loc('OPS_IS_UNIX'),
>                            path => 'Search/Results.html?Query=Queue=\'OPS_IS_UNIX\' AND
(Status = \'open\' OR Status = \'new\' OR Status = \'stalled\')&Rows=50'
>                          },
>                     ZR => { title => loc('OPS_IS_Windows'),
>                            path => 'Search/Results.html?Query=Queue=\'OPS_IS_Windows\'
AND (Status = \'open\' OR Status = \'new\' OR Status = \'stalled\')&Rows=50'
>                          },
>
```

/opt/rt3/share/html/Elements/Tabs (Request Tracker V3.6.3 only)

```
68,70c68,70
<                     Ab => { title => loc('Simple Search'),
<                         path => 'Search/Simple.html'
<                          },
---
> #                    Ab => { title => loc('Simple Search'),
> #                        path => 'Search/Simple.html'
> #                         },
72c72
<                         path => 'Search/Build.html'
---
>                        path => 'Search/Build.html'
81a82,102
> # Generate a list of queues which the user has access to. This capability will allow
the user
> # to quickly switch between queues, if they have to work within or manage multiple
queues.
>
>     my $q = new RT::Queues($session{'CurrentUser'});
>     $q->UnLimit;
>     my $counter = 1;
>     my @subchar = qw(AA AB AC AD AE AF AG AH AI AJ AK AL AM AN AO AP AQ AR AS AT AU AV
AW AX AY AZ BA BB BC BD BE BF BG BH BI BJ BK BL BM BN BO BP BQ BR BS BT BU BV BW BX BY
BZ);
>     my $subval;
>     while (my $queue = $q->Next) {
>         if ($queue->CurrentUserHasRight('SeeQueue')) {
>             my $queue_name = $queue->Name;
>              $subval = "Z".$subchar[$counter];
>              $basetabs->{$subval} = {
>                 title => loc("$queue_name"),
>                 path => "Search/Results.html?Query=Queue=\'$queue_name\' AND (Status
= \'open\' OR Status = \'new\' OR Status = \'stalled\')&Rows=50"
>              };
>              $counter++;
>         }
>     }
>
>
```

For Request Tracker V3.6.3 only: Comment out lines 68 – 70 which activates the **SimpleSearch** menu item. **SimpleSearch** has the capability to search the text of all tickets; however, the search can have a negative impact on Request Tracker and host system performance, if this feature is available.

/opt/rt3/share/html/NoAuth/webrt.css (Request Tracker V3.4.4 only)

```
47c47
<       font-size: 12px;
---
>       font-size: 13px;
51c51
< .nav2 {       font-size: 10px;
---
> .nav2 {       font-size: 12px;
55c55
<       font-weight: normal;
---
>       font-weight: bold;
66,67c66,67
<       font-size: 16px;
<       font-weight: normal;
---
>       font-size: 13px;
>       font-weight: bold;
78,79c78,79
<       font-size: 16px;
<       font-weight: normal;
---
>       font-size: 13px;
>       font-weight: vold;
84,85c84,85
<       font-size: 14px;
<       font-weight: normal;
---
>       font-size: 12px;
>       font-weight: bold;
90,91c90,91
<       font-size: 12px;
<       font-weight: normal;
---
>       font-size: 11px;
>       font-weight: bold;
97c97
<       font-weight: normal;
---
>       font-weight: bold;
103c103
<       font-weight: normal;
---
>       font-weight: bold;
109c109
<       font-weight: normal;
---
>       font-weight: bold;
114,116c114,116
<       border-top: solid #999999 1px;
<       padding-top: .1em;
<       margin-top: .5em;
---
>       border-top: none;
>       padding-top: .0em;
>       margin-top: .0em;
119,121c119,121
<       border-top: solid #999999 1px;
<       padding-top: .1em;
<       margin-top: .5em;
---
>       border-top: none;
>       padding-top: .0em;
>       margin-top: .0em;
124,126c124,126
<       border-top: solid #999999 1px;
<       padding-top: .1em;
<       margin-top: .5em;
---
```

```
>       border-top: none;
>       padding-top: .0em;
>       margin-top: .0em;
129,131c129,131
<       border-top: solid #999999 1px;
<       padding-top: .1em;
<       margin-top: .5em;
---
>       border-top: none;
>       padding-top: .0em;
>       margin-top: .0em;
134,136c134,136
<       border-top: solid #999999 1px;
<       padding-top: .1em;
<       margin-top: .5em;
---
>       border-top: none;
>       padding-top: .0em;
>       margin-top: .0em;
139,141c139,141
<       border-top: solid #999999 1px;
<       padding-top: .1em;
<       margin-top: .5em;
---
>       border-top: none;
>       padding-top: .0em;
>       margin-top: .0em;
144,146c144,146
<       border-bottom: solid white 1px;
<       padding-top: .25em;
<       padding-bottom: .5em;
---
>       border-bottom: none;
>       padding-top: .0em;
>       padding-bottom: .0em;
149,151c149,151
<       border-bottom: solid white 1px;
<       padding-top: .25em;
<       padding-bottom: .5em;
---
>       border-bottom: none;
>       padding-top: .0em;
>       padding-bottom: .0em;
154,156c154,156
<       border-bottom: solid white 1px;
<       padding-top: .25em;
<       padding-bottom: .5em;
---
>       border-bottom: none;
>       padding-top: .0em;
>       padding-bottom: .0em;
159,161c159,161
<       border-bottom: solid white 1px;
<       padding-top: .25em;
<       padding-bottom: .5em;
---
>       border-bottom: none;
>       padding-top: .0em;
>       padding-bottom: .0em;
164,166c164,166
<       border-bottom: solid white 1px;
<       padding-top: .25em;
<       padding-bottom: .5em;
---
>       border-bottom: none;
>       padding-top: .0em;
>       padding-bottom: .0em;
169,171c169,171
<       border-bottom: solid white 1px;
<       padding-top: .25em;
<       padding-bottom: .5em;
```

```
---
>       border-bottom: none;
>       padding-top: .0em;
>       padding-bottom: .0em;
175c175
<       font-size: 16px;
---
>       font-size: 13px;
187c187
<       font-size: 16px;
---
>       font-size: 13px;
193c193
<       font-size: 14px;
---
>       font-size: 12px;
199,200c199,200
<       font-size: 12px;
<         font-weight: normal;
---
>       font-size: 11px;
>         font-weight: bold;
206c206
<         font-weight: normal;
---
>         font-weight: bold;
212c212
<         font-weight: normal;
---
>         font-weight: bold;
218c218
<         font-weight: normal;
---
>         font-weight: bold;
223,225c223,225
<       border-top: solid #999999 1px;
<       padding-top: .1em;
<       margin-top: .5em;
---
>       border-top: none;
>       padding-top: .0em;
>       margin-top: .0em;
228,230c228,230
<       border-top: solid #999999 1px;
<       padding-top: .1em;
<       margin-top: .5em;
---
>       border-top: none;
>       padding-top: .0em;
>       margin-top: .0em;
233,235c233,235
<       border-top: solid #999999 1px;
<       padding-top: .1em;
<       margin-top: .5em;
---
>       border-top: none;
>       padding-top: .0em;
>       margin-top: .0em;
238,240c238,240
<       border-top: solid #999999 1px;
<       padding-top: .1em;
<       margin-top: .5em;
---
>       border-top: none;
>       padding-top: .0em;
>       margin-top: .0em;
243,245c243,245
<       border-top: solid #999999 1px;
<       padding-top: .1em;
<       margin-top: .5em;
---
```

```
>       border-top: none;
>       padding-top: .0em;
>       margin-top: .0em;
248,250c248,250
<       border-top: solid #999999 1px;
<       padding-top: .1em;
<       margin-top: .5em;
---
>       border-top: none;
>       padding-top: .0em;
>       margin-top: .0em;
253,255c253,255
<       border-bottom: solid white 1px;
<       padding-top: .25em;
<       padding-bottom: .5em;
---
>       border-bottom: none;
>       padding-top: .0em;
>       padding-bottom: .0em;
258,260c258,260
<       border-bottom: solid white 1px;
<       padding-top: .25em;
<       padding-bottom: .5em;
---
>       border-bottom: none;
>       padding-top: .0em;
>       padding-bottom: .0em;
263,265c263,265
<       border-bottom: solid white 1px;
<       padding-top: .25em;
<       padding-bottom: .5em;
---
>       border-bottom: none;
>       padding-top: .0em;
>       padding-bottom: .0em;
268,270c268,270
<       border-bottom: solid white 1px;
<       padding-top: .25em;
<       padding-bottom: .5em;
---
>       border-bottom: none;
>       padding-top: .0em;
>       padding-bottom: .0em;
273,275c273,275
<       border-bottom: solid white 1px;
<       padding-top: .25em;
<       padding-bottom: .5em;
---
>       border-bottom: none;
>       padding-top: .0em;
>       padding-bottom: .0em;
278,280c278,280
<       border-bottom: solid white 1px;
<       padding-top: .25em;
<       padding-bottom: .5em;
---
>       border-bottom: none;
>       padding-top: .0em;
>       padding-bottom: .0em;
284c284
<       font-size: 10px;
---
>       font-size: 9px;
289,290c289,290
<       font-size: 11px;
<       font-weight: normal;
---
>       font-size: 8px;
>       font-weight: bold;
295c295
<       font-size: 11px;
```

```
---
>       font-size: 8px;
```

/opt/rt3/share/html/NoAuth/css/3.4-compat/nav.css (Request Tracker V3.6.3 only)

```
50c50
<     font-size: 1.4em;
---
>     font-size: 1.0em;
58a59
>     font-weight: bold;
65a67
>     font-weight: bold;
69a72
>     font-weight: bold;
73,74c76,77
<     padding: 0.4em 0 0.4em 0.2em;
<     border-bottom: 1px solid white;
---
>     padding: 0.0em 0 0.0em 0.0em;
>     border-bottom: 0px solid white;
78c81,82
<     font-size: 0.9em;
---
>     font-size: 1.0em;
>     font-weight: bold;
```

Once the above files have been modified, then do the following:

```
/usr/local/apache2/bin/apachectl stop
rm -rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

4.8. Additional “Save Changes” capability/Custom Field Validation

In order to improve efficiency, a couple minor modifications were made to **/opt/rt3/share/html/Ticket/Modify.html, /opt/rt3/share/html/Ticket/ModifyAll.html** and **/opt/rt3/share/html/Ticket/Create.html.** These changes place an addition Save Changes submit button at the top of each page.

Also, a patch was added to **/opt/rt3/share/html/Tcket/Modify.html** and **/opt/rt3/share/html/Ticket/ModifyAll.html**, in Request Tracker V3.6.3, to “flag” custom field validation. The patch also generates error message to force the user to enter information in custom fields, it they are mandatory.

/opt/rt3/share/html/Ticket/Modify.html

Request Tracker V3.4.4:

```
54a55,60
> <& /Elements/Submit,
>     Label => loc('Save Changes'),
>     Caption => loc("If you've updated anything below, be sure to"), color => "#333399"
&>
>
> <br>
>
```

Request Tracker V3.6.3:

```
49a50
> </form>
54a56
> <& /Elements/Submit, Label => loc('Save Changes'), Caption => loc("If you've updated
anything below, be sure to"), color => "#993333" &>
63c65
<
---
>
69d70
<
73a75,104
> # Add validation
>
> my $ValidCFs = $m->comp(
>     '/Elements/ValidateCustomFields',
>     CustomFields => $CustomFields,
>     ARGSRef => \%ARGS
> );
>
> if (!exists $ARGS{'AddMoreAttach'}) {
>
>     if ($ValidCFs) {
>         $m->comp('Display.html', %ARGS);
>         $RT::Logger->crit("After display call; error is $@");
>         $m->abort();
>     }
>     else {
>         # Invalid CFs
>         while (my $CustomFields = $CustomFields->Next) {
>           my $cfn = $CustomFields->Name;
>           my $cfv = $TicketObj->FirstCustomFieldValue($cfn);
>             if ($cfv eq "") {
>                     my $msg = $m->notes('InvalidField-' . $CustomFields->Id) or next;
>                     push @results, $CustomFields->Name . ': ' . $msg;
>             }
>         }
>     }
> }
>
> # End validation
>
```

/opt/rt3/share/html/Ticket/ModifyAll.html

Request Tracker V3.4.4:

```
57a58,63
> <& /Elements/Submit,
>     Label => loc('Save Changes'),
>     Caption => loc("If you've updated anything below, be sure to"), color => "#333399"
&>
>
> <BR>
>
```

Request Tracker V3.6.3:

```
57a58,63
> <& /Elements/Submit,
>     Label => loc('Save Changes'),
>     Caption => loc("If you've updated anything below, be sure to"), color => "#333399"
&>
>
> <BR>
>
136a143,154
> # Add validation
>
> my $CustomFields = $Ticket->QueueObj->TicketCustomFields();
>
> my $ValidCFs = $m->comp(
>       '/Elements/ValidateCustomFields',
>       CustomFields => $CustomFields,
>       ARGSRef => \%ARGS
> );
>
> # End Validation
>
190a209,233
>
> # Add validation
>
> if (!exists $ARGS{'AddMoreAttach'}) {
>
>       if ($ValidCFs) {
>           $m->comp('Display.html', %ARGS);
>           $RT::Logger->crit("After display call; error is $@");
>           $m->abort();
>       }
>       else {
>          # Invalid CFs
>          while (my $CustomFields = $CustomFields->Next) {
>             my $cfn = $CustomFields->Name;
>             my $cfv = $Ticket->FirstCustomFieldValue($cfn);
>             if ($cfv eq "") {
>                    my $msg = $m->notes('InvalidField-' . $CustomFields->Id) or next;
>                    push @results, $CustomFields->Name . ': ' . $msg;
>             }
>         }
>     }
> }
>
```

```
> # End validation
>
```

/opt/rt3/share/html/Ticket/Create.html

```
57a58          (For Request Tracker V3.6.3: 60a61)
> <& Elements/Submit, Label => loc(“Create”)&>
```

Once the above files have been modified, then do the following:

```
/usr/local/apache2/bin/apachectl stop
rm –rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

4.9. Making Time Worked a mandatory field upon Resolve

The default behavior of Request Tracker is to allow users to resolve tickets, without having to enter a value in the **Time Worked** field. While **Time Worked** can be filled in at each stage of the ticket process, i.,e, Reply To, Comment On or Resolve, this patch will force the user to enter a value in **Time Worked**, before the ticket is resolved.

The following is applicable to Request Tracker V3.6.3:

/opt/rt3/share/html/Ticket/Update.html

```
156a157,160
>     if ((! defined($TicketObj->TimeWorked) or $TicketObj->TimeWorked == 0) and
>       $ARGS{'UpdateTimeWorked'} <= 0 and exists $ARGS{'SubmitTicket'} ) {
>      Abort("You cannot resolve a ticket with zero Time Worked. Please use the back
arrow and enter a value in Time Worked, then click on Update Ticket");
>     }
```

/opt/rt3/share/html/Ticket/Elements/EditBaiscs

```
79,80c79,81
<       <td class="value"><input name="TimeWorked" value="<%$TicketObj->TimeWorked|h%>"
size="5" />
<     <& /Elements/SelectTimeUnits, Name =>'TimeWorked' &>
---
>       <td class="value"><%$TicketObj->TimeWorked|h%> Minutes
> %#      <td class="value"><input name="TimeWorked" value="<%$TicketObj->TimeWorked|h%>"
size="5" />
> %#    <& /Elements/SelectTimeUnits, Name =>'TimeWorked' &>
```

Note: The above change prevents the user from overriding the **Time Worked** field**.** Because the user can enter a value in **Time Worked** at various points of the Ticket lifecycle, this field needed to be deactivated in this script so the accumulated Time Worked value would not be accidentally wiped out or altered in any way.

4.10. Restrict “newest unowned tickets” display

The default behavior of Request Tracker is to display tickets from all the queues available, which makes it difficult to locate new tickets of interest for the Request Tracker user. The following small patch will enable Request Tracker to display tickets to the user related to the queue or queues they service.

The following is applicable to Request Tracker V3.4.4:

The **/opt/rt3/share/html/Elements/MyRequests** file needs to be modified as follows:

```
62c62,78
< my $Query = "Owner = 'Nobody' AND ( Status = 'new' OR Status = 'open')";
---
> # The next section of code will limit unowned tickets to only be
> # those that are in a queue that a user can own tickets in
>
> my $q = new RT::Queues($session{'CurrentUser'});
> $q->UnLimit;
> my @queues;
>
> while (my $queue = $q->Next) {
>       if ($queue->CurrentUserHasRight( 'OwnTicket' )) {
>               push( @queues, "Queue = \'".$queue->Name."\'" );
>       }
> }
>
> my $Query = "Owner = 'Nobody' AND ( Status = 'new' OR Status = 'open')
> AND (".join( " OR ",@queues ).")";
>
> # End code section
```

The following is applicable to Request Tracker V3.6.3

The **/opt/rt3/share/html/Elements/ShowSearch** file needs to be modified as follows:

```
114c114,140
< my $QueryString = '?' . $m->comp( '/Elements/QueryString', %$SearchArg );
---
> my $QueryString;
>
> if ($Name eq "Unowned Tickets") {
>
>       # The next section of code will limit unowned tickets to only be
>       # those that are in a queue that a user can own tickets in
>
>       my $q = new RT::Queues($session{'CurrentUser'});
>       $q->UnLimit;
>       my @queues;
>
>       while (my $queue = $q->Next) {
>               if ($queue->CurrentUserHasRight( 'OwnTicket' )) {
>                       push( @queues, "Queue = \'".$queue->Name."\'" );
>               }
>       }
>
>       my $unowned_queues = " AND "."(".join( " OR ",@queues ).")";
>       my $SavedQuery = $SearchArg->{'Query'};
>       $SearchArg->{'Query'} = $SavedQuery.$unowned_queues;
>
>       # End code section
>
>       $QueryString = '?' . $m->comp( '/Elements/QueryString', %$SearchArg );
> } else {
>       $QueryString = '?' . $m->comp( '/Elements/QueryString', %$SearchArg );
> }
```

Once the above files have been modified, then do the following:

```
/usr/local/apache2/bin/apachectl stop
rm -rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

4.11. Drop Down List Window Size Increase

The Drop Down List, pick list, display area by default is set to 5 lines, unfortunately, long Drop Down Lists are harder to use when only 5 lines are displayed. This change increases the line display to 10 lines. One unfortunately side affect is that this change is universal, as Drop Down List line display can not be set on a field by field basis.

/opt/rt3/share/html/Elements/EditCustomField

```
86c86        (For Request Tracker V3.6.3: 94c94)
< $Rows = 5
---
> $Rows = 10
```

Once the above files have been modified, then do the following:

```
/usr/local/apache2/bin/apachectl stop
rm -rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

4.12. The guest Account User Menu

Request Tracker displays an automatically generated menu, or the **New Ticket** page, which enables a **guest** account user to enter Tickets. Unfortunately, what is displayed back to the user is not descriptive enough for the end user. In addition, there are issues with tickets not being successfully created by using the Request Tracker generated interface. With these issues, it was decided to replace the Request Tracker generated interface with external CGI scripts. The CGI scripts will collect the information from the user, have the capability of required fields, and can e-mail tickets directly to Request Tracker, thus eliminating the unsuccessful ticket creation issue.

This section describes how to implement the CGI scripts into the Request Tracker environment. It does not go into the details of creating the CGI scripts themselves. However, the CGI scripts are well documented and are available on the Request Tracker production system (helpdesk.digitalglobe.com) in **/usr/local/apache2/cgi-bin**. The scripts are as follows:

Script Name	Description
rt_bs_menu.cgi	Business Systems
rt_ct_menu.cgi	Commercial Technical Support
rt_dt_menu.cgi	Defense Technical Support
rt_fac_menu.cgi	Facilities Systems
rt_gps_menu.cgi	GeoOps Production Services
rt_gspg_menu.cgi	GeoOps SPG
rt_is_menu.cgi	Information Systems
rt_phx_menu.cgi	Information Systems - Phoenix
rt_walc_menu.cgi	Information Systems – Walnut Creek
rt_ps_menu.cgi	Program Security
rt_train_menu.cgi	Training

4.12.1. Creating the Customized Menu Web Page

The procedure to create the customized web page is relatively simple.

First save a copy of Request Tracker’s generated web page:

```
cd /opt/rt3/share/html/SelfService
cp CreateTicketInQueue.html CreateTicketInQueue.html.dist
```

Now, edit **CreateTicketInQueue.html** and comment out lines 51 – 56, by placing “**%#**” in front of each line. This code is used to automatic generation.

Next, for each queue desired, do the following:

```
cd /opt/rt3/share/html/SelfService
```

Edit CreateTicketInQueue.html, as follows:

```
<dt><a=href=”http://helpdesk.digitalglobe.com:81/cgi-bin/name of cgi.cgi”>Name
of Queue</a></dt>
<dd>Description</dd>
```

The following is an example of a CGI script added to the menu:

```
<dt><a=href=”http://helpdesk.digitalglobe.com:81/cgi-
bin/rt_is_menu.cgi”>Information Systems</a></dt>
<dd>Requestes related to Information Systems</dd>
```

The following is a complete copy of the production version of CreateTicketInQueue.html:

```
%# BEGIN BPS TAGGED BLOCK {{{
%# 
%# COPYRIGHT:
%#  
%# This software is Copyright (c) 1996-2005 Best Practical Solutions, LLC 
%#                                          <jesse@bestpractical.com>
%# 
%# (Except where explicitly superseded by other copyright notices)
%# 
%# 
%# LICENSE:
%# 
%# This work is made available to you under the terms of Version 2 of
%# the GNU General Public License. A copy of that license should have
%# been provided with this software, but in any event can be snarfed
%# from www.gnu.org.
%# 
%# This work is distributed in the hope that it will be useful, but
%# WITHOUT ANY WARRANTY; without even the implied warranty of
%# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
%# General Public License for more details.
%# 
%# You should have received a copy of the GNU General Public License
%# along with this program; if not, write to the Free Software
%# Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
%# 
%# 
%# CONTRIBUTION SUBMISSION POLICY:
%# 
%# (The following paragraph is not intended to limit the rights granted
%# to you to modify and distribute this software under the terms of
%# the GNU General Public License and is only of importance to you if
%# you choose to contribute your changes and enhancements to the
%# community by submitting them to Best Practical Solutions, LLC.)
%# 
%# By intentionally submitting any modifications, corrections or
%# derivatives to this work, or any other work intended for use with
%# Request Tracker, to Best Practical Solutions, LLC, you confirm that
%# you are the copyright holder for those contributions and you grant
%# Best Practical Solutions,  LLC a nonexclusive, worldwide, irrevocable,
%# royalty-free, perpetual, license to use, copy, create derivative
%# works based on those contributions, and sublicense and distribute
%# those contributions and any derivatives thereof.
%# 
%# END BPS TAGGED BLOCK }}}
<& Elements/Header, Title => loc("Create a ticket") &>

<h1><&|/l&>Select a queue for your new ticket</&></h1>

<dl>
%# while (my $queue = $queues->Next) {
%# next unless $queue->CurrentUserHasRight('CreateTicket');

%#<dt><a href="<%$RT::WebPath%>/SelfService/Create.html?Queue=<%$queue->id%>"><%$queue-
>Name%></a></dt>
%#<dd><%$queue->Description%></dd>
%# }

<b><h3>DigitalGlobe Departmental Help Desk Queues</h3></b>
<p>
```

```
<dt><a href="http://helpdesk.digitalglobe.com:81/cgi-bin/rt_bs_menu.cgi">Business
Systems</a></dt>
<dd>Requests related to DigitalGlobe Business Systems Support, e.g. SAP</dd>
<dt><a href="http://helpdesk.digitalglobe.com:81/cgi-bin/rt_ct_menu.cgi">Commercial
Technical Support</a></dt>
<dd>Request related to Commercial Technical Support Requests</dd>
<dt><a href="http://helpdesk.digitalglobe.com:81/cgi-bin/rt_dt_menu.cgi">Defense
Technical Support</a></dt>
<dd>Request related to Defense Technical Support Requests</dd>
<dt><a href="http://helpdesk.digitalglobe.com:81/cgi-
bin/rt_fac_menu.cgi">Facilities</a></dt>
<dd>Requests related to DigitalGlobe facilities</dd>
<dt><a href="http://helpdesk.digitalglobe.com:81/cgi-bin/rt_gps_menu.cgi">GeoOps
ProdSvc</a></dt>
<dd>This area supports Geospatial Operations Production Service Requests (Feasibility
Requests,
RQI, Coproduction Source Packaging, etc).</dd>
<dt><a href="http://helpdesk.digitalglobe.com:81/cgi-bin/rt_gspg_menu.cgi">GeoOps
SPG</a></dt>
<dd>The SPG group supports Standard Order Status Requests. This request tool replaces the
current Geo Ops Service Request Form for the following types of requests: Order Status
Inquiry,
Force Production, Strip Catalog ID Creation, Media Change, Cloud Cover Reassessment, Rush
Archive Order,
Priority Increases, and Fragment Modification. </dd>
<dt><a href="http://helpdesk.digitalglobe.com:81/cgi-bin/rt_is_menu.cgi">Information
Systems</a></dt>
<dd>Requests related to Information Systems</dd>
<dt><a href="http://helpdesk.digitalglobe.com:81/cgi-bin/rt_phx_menu.cgi">Information
Systems</a></dt>
<dd>Requests related to Information Systems - Phoenix</dd>
<dt><a href="http://helpdesk.digitalglobe.com:81/cgi-bin/rt_walc_menu.cgi">Information
Systems</a></dt>
<dd>Requests related to Information Systems – Walnut Creek</dd>
<dt><a href="http://helpdesk.digitalglobe.com:81/cgi-bin/rt_ps_menu.cgi">Program
Security</a></dt>
<dd>Request related to Program Security</dd>
</dl>
<HR>
<b>NOTICE:</b> If you are using any portion of the Request Tracker Suite from a
laptop PC, and you plan to undock and/or move the laptop to a different location, please
logoff from Request Tracker BEFORE relocating the laptop. Under certain
circumstances, Request Tracker may not function properly as network connectivity may
be lost during the undock/move and this could cause a database connectivity error.
<HR>
<%init>
my $queues = RT::Queues->new($session{'CurrentUser'});
$queues->UnLimit;
</%init>
```

NOTICE: Please see the section: *Adding a notification concerning laptop computers* concerning a notice which needs to be added to this file.

Once all the queues have been entered, then do the following:

```
/usr/local/apache2/bin/apachectl stop
rm –rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

4.12.2. Customize the guest Account User Interface

The default Request Tracker set up has the user logging into the **guest** account, and directing the user to the **Open Tickets** page. It is desired that the user is directed to the **New Tickets** page instead (please see previous section). It is a simple change to make the **New Tickets** page the first page the user sees when the log in. This is accomplished by doing the following:

```
cd /opt/rt3/share/html/SelfService
mv index.html index.html.keep
ln -s index.html.kep Open.html
ln -s CreateTicketInQueue.html index.html
chown -h apache:rt *.html
```

Next, `cd /opt/rt3/share/html/SelfService:`

Edit Tabs, as follows:

```
64c64
<                        path => 'SelfService/',
---
>                        path => 'SelfService/Open.html',
73,78d72
< if ($session{'CurrentUser'}->HasRight( Right => 'ModifySelf',
<                                     Object => $RT::System )) {
<       $tabs->{Z} = { title => loc('Preferences'),
<                     path => 'SelfService/Prefs.html'
<                     };
< }
```

The above change redirects the Open Ticket menu item to **Open.html**; the link created above. The second change removes the capability for the user to change the **guest** account password.

Once all the changes have been made, then do the following:

```
/usr/local/apache2/bin/apachectl stop
rm -rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

4.12.3. Create the guest Account

Log into Request Tracker and create the **guest** account, as follows:

Configuration -> **Users** -> **New User**

Enter **guest** for the user name, **CHANGME@digitalglobe.com** for the e-mail address and **guest** for the password.

This is all that is required

4.12.4. Using the guest Account

To access the **guest** account via a web browser:

```
http://helpdesk.digitalglobe.com/SelfService/?user=guest&pass=guest
```

The above URL will take the user to the desired location to select a **Queue** in order to enter a ticket.

4.12.5. Adding “stalled” tickets to various ticket displays

In Request Tracker V3.6.3, the authors have suppressed the display of “stalled” tickets in the various ticket displays. While they returned “stalled” tickets to the “RT at a glance” screen, they removed this capability from various ticket displays. As DigitalGlobe used the “stalled” tickets capability, several files had to be modified to include the “stalled” state.

So, in the following files, please add “`OR Status = ‘stalled’`” after “`Status = ‘new’`:

/opt/rt3/share/html/Ticket/Elements/ShowRequestor
/opt/rt3/share/html/Elements/MyAdminQueues
/opt/rt3/share/html/Elements/MyReminders
/opt/rt3/share/html/Elements/MySupportQueues

4.13. Request Tracker V3 Custom Fields

4.13.1. Setting up Custom Fields

Custom Fields can be set up using the **Configuration** menu item on the Request Tracker Menu, as follows:

Configuration -> **Custom Fields** -> **New custom field**

4.13.2. Activating Custom Fields for Usage

To activate a **Custom Field**, so it will be displayed on **Tickets** and the **Ticket Query Builder**, do the following:

Configuration -> **Global** -> **Custom Fields** -> **Tickets**

Click on the box next to the desired fields and than click on **Submit**.

This screen provides the option to reorder **Custom Fields** displayed on the **Ticket**, by selecting **Move up** and **Move down**, next to each **Custom Field**.

4.13.3. Activating Custom Fields on Queues

In addition to activating custom fields on a Global basis, custom fields need to be activated on a **Queue** basis, as well. This can be done by doing the following:

Configuration -> **Custom Fields** -> **Custom Field Name** -> **Applies to**

Select the **Queue(s)** to which the **Custom Field** applies to.

4.13.4. Activating Custom Fields for the guest User Account

Due to the limited privileges on the **guest** user account, additional steps need to be taken so **Custom Fields** and **Queues**, will be displayed properly when a user uses the **guest** user account. For each **Queue** desired, do the following:

Configuration -> **Queues** -> Desired Queue

Then:

Select **Group Rights** and assign right **CreateTicket,** to the **Everyone** and **Unprivileged** groups. This set up applies to the **Queue** to which users will submit requests, e.g. **A_General**, and gives them the capability of creating and viewing Tickets.

For other queues, e.g. **A_Projects**, the **guest** account will only have view ticket and view queue access. This is done, as follows:

Configuration -> **Global** -> **Custom Fields** -> **Tickets** -> **Group Rights**

Assign rights **SeeQueue, ShowTicket, ShowTicketComments** and **Watch** to the **Everyone** and **Unprivileged** groups.

For each **Custom Field**, which is desired to be displayed on the **Ticket** being entered, do the following:

Configuration -> **Custom Fields** -> Custom Field -> **Group Rights**

Assign rights **ModifyCustomField** and **SeeCustomField** to the **Everyone** and **Unprivileged** groups.

4.14. Request Tracker Ticket E-mail

4.14.1. Introduction

The Request Tracker software provides a set of templates which are used to format correspondence for e-mail messages. The templates a rather generic in nature and did not provide enough information to the user community. This section provides detailed information on the customizations made to the various templates. Therefore, when an upgrade is performed, the templates need to be updated accordingly.

4.14.2. Update Templates

The procedure to update a template is as follows:

Configuration -> **Global** -> **Templates**

Then:

Select the desired Template by clicking on the name in **bold**.

Modify the template as desired and click on **Save changes**.

Once the changes are save, the new template takes affect immediately.

In the next few pages are a copy of the updated templates and their contents. This information is provided to reconstruct the templates, in the event a template is removed by an upgrade or by accident. These sections are named by the template name.

4.14.3. Autoreply

```
Subject: AutoReply: {$Ticket->Subject}


Greetings,

This message has been automatically generated in response to the
creation of a trouble ticket regarding:
        "{$Ticket->Subject()}",
a summary of which appears below.

There is no need to reply to this message right now.  Your ticket has been
assigned an ID of [{$rtname} #{$Ticket->id()}].

Please include the string:

         [{$rtname} #{$Ticket->id}]

in the subject line of all future correspondence about this issue. You may reply to this
message.

----------------------------------------------------------------------------------
{$Transaction->Content()}

----------------------------------------------------------------------------------
To review your ticket at any time, please go to:

<URL: {$RT::WebURL}SelfService/Display.html?id={$Ticket->id}&user=guest&pass=guest >
```

4.14.4. Transaction

```
RT-Attach-Message: yes


{$Transaction->CreatedAsString}: Request {$Ticket->id} was acted upon.
Transaction: {$Transaction->Description}
       Queue: {$Ticket->QueueObj->Name}
     Subject: {$Transaction->Subject || $Ticket->Subject || "(No subject given)"}
       Owner: {$Ticket->OwnerObj->Name}
  Requestors: {$Ticket->RequestorAddresses}
      Status: {$Ticket->Status}
 Ticket <URL: {$RT::WebURL}Ticket/Display.html?id={$Ticket->id} >


{$Transaction->Content()}

----------------------------------------------------------------------------------
To review your ticket at any time, please go to:

<URL: {$RT::WebURL}SelfService/Display.html?id={$Ticket->id}&user=guest&pass=guest >
```

4.14.5. Admin Correspondence

```
RT-Attach-Message: yes


<URL: {$RT::WebURL}Ticket/Display.html?id={$Ticket->id} >

{$Transaction->Content()}

----------------------------------------------------------------------------------
To review your ticket at any time, please go to:

<URL: {$RT::WebURL}SelfService/Display.html?id={$Ticket->id}&user=guest&pass=guest >
```

4.14.6. Correspondence

```
RT-Attach-Message: yes

{$Transaction->Content()}

----------------------------------------------------------------------------------
To review your ticket at any time, please go to:

<URL: {$RT::WebURL}SelfService/Display.html?id={$Ticket->id}&user=guest&pass=guest >
```

4.14.7. Admin Comment

```
Subject: [Comment] {my $s=($Transaction->Subject||$Ticket->Subject); $s =~ s/\
[Comment\]//g; $comment =~ s/^Re//i; $s;}


{$RT::WebURL}Ticket/Display.html?id={$Ticket->id}
This is a comment.  It is not sent to the Requestor(s):

{$Transaction->Content()}

----------------------------------------------------------------------------------
To review your ticket at any time, please go to:

<URL: {$RT::WebURL}SelfService/Display.html?id={$Ticket->id}&user=guest&pass=guest >
```

4.14.8. Status Change

```
Subject: Status Changed to: {$Transaction->NewValue}


{$RT::WebURL}Ticket/Display.html?id={$Ticket->id}

{$Transaction->Content()}

-------------------------------------------------------------------------------
To review your ticket at any time, please go to:

<URL: {$RT::WebURL}SelfService/Display.html?id={$Ticket->id}&user=guest&pass=guest >
```

4.14.9. Resolved

```
Subject: Resolved: {$Ticket->Subject}

*** PLEASE DO NOT RESPOND TO THIS MESSAGE, UNLESS THE RESOLUTION DOES NOT MEET WITH YOUR
SATISFACTION ***

According to our records, your request has been resolved. If you desire additional help
regarding this subject, please respond to this message. If not, we’re glad to have been
of assistance and you need not send a “thank you” reply.

----------------------------------------------------------------------------------
RT-Attach-Message: yes

{$Transaction->CreatedAsString}: Request {$Ticket->id} was acted upon.
Transaction: {$Transaction->Description}
       Queue: {$Ticket->QueueObj->Name}
     Subject: {$Transaction->Subject || $Ticket->Subject || "(No subject given)"}
       Owner: {$Ticket->OwnerObj->Name}
  Requestors: {$Ticket->RequestorAddresses}
      Status: {$Ticket->Status}
 Ticket <URL: {$RT::WebURL}SelfService/Display.html?id={$Ticket-
>id}&user=guest&pass=guest >

To review your Ticket, please click on the link above.
```

4.14.10. Owner Change Tell Requestor

```
RT-Attach-Message: yes


{$Transaction->CreatedAsString}: Request {$Ticket->id} was acted upon.
Transaction: {$Transaction->Description}
       Queue: {$Ticket->QueueObj->Name}
     Subject: {$Transaction->Subject || $Ticket->Subject || "(No subject given)"}
       Owner: {$Ticket->OwnerObj->Name}
  Requestors: {$Ticket->RequestorAddresses}
      Status: {$Ticket->Status}
 Ticket <URL: {$RT::WebURL}Ticket/Display.html?id={$Ticket->id} >

Your ticket has been changed to a new owner as noted above.

{$Transaction->Content()}

----------------------------------------------------------------------------------
To review your ticket at any time, please go to:

<URL: {$RT::WebURL}SelfService/Display.html?id={$Ticket->id}&user=guest&pass=guest >
```

4.14.11. Update Scrips

This section presents three new scripts. One will send e-mail when a ticket is moved from one queue to another. The second scrip will send an e-mail to the new ticket owner when the ownership of the ticket is changed to a new owner. The third scrip will send an e-mail to the requestor when the ownership of the ticket is changed to a new owner.

The procedure to add a new scrip is as follows:

Configuration -> **Global** -> **Scrips** -> **New Scrip**

Then:

Select the desired information as shown below and click on **Submit**.

Once the changes are save, the new scrip takes affect immediately.

On the next few lines are a copy of the new scrips and their contents. This information is provided to reconstruct the scrips, in the event a scrip is removed by an upgrade or by accident. These sections are named by the scrip name.

4.12.3.1. Queue Change scrip

```
Condition: On Queue Change
Action: Notify AdminCcs
Template: Global template: Transaction
Stage: TransactionCreate
```

4.12.3.2. Owner Change scrip

```
Condition: On Owner Change
Action: Notify Owner
Template: Global template: Transaction
Stage: TransactionCreate
```

4.12.3.3. Owner Change Tell Requestor scrip

```
Condition: On Owner Change
Action: Notify Requestors
Template: Global template: Owner Change
Stage: TransactionCreate
```

5. Conclusion

DigitalGlobe Information Systems has purchased **RT Essentials**, which is authored by Jesse Vincent, Robert Spier, Dave Rolsky, Darren Chamberlain and Richard Foley; it is published by O;Reilly. This book provides supplemental information about installing Request Tracker and the setting up and maintenance of the Request Tracker environment. It is highly recommended that responsible System Administration staff become familiar with the contents of this book.

6. Appendix

6.1.1. Sendmail – SMTP AUTH Setup on RedHat Enterprise

Edit the sendmail.mc file

Use the "cd" command to go to the /etc/mail directory, and use your favorite text editor to open the **sendmail.mc** file. Look for these three lines somewhere in there:

```
dnl define(`confAUTH_OPTIONS', `A')dnl
dnl TRUST_AUTH_MECH(`DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
dnl define(`confAUTH_MECHANISMS', `DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
```

Notice that two of those lines start with the letters "dnl". That means "delete through newline" and is sendmail's way of either ending a line or starting a comment. You need to eliminate the "dnl" at the start of the line in order to activate the functionality in these lines. When you're done, the three lines should look exactly like this:

```
define(`confAUTH_OPTIONS', `A')dnl
TRUST_AUTH_MECH(`DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
```

WARNING: Those are *directed quotes*! They are not single quotes, double quotes, or smart quotes. You type the left directed quote with the backtick (also known as the "accent grave" key by the French and usually located on the same physical key as the ~ character), and you type the right directed quote with the apostrophe.

On some systems, the first of those three lines will show **`A p'** instead of just **`A'.** That additional "p" letter is to disable plaintext SMTP AUTH if the communications session is not encrypted, to avoid sending out the user's mail username and password in the clear. However, both the POP and IMAP services used for checking mail already do send that password without encryption, so by removing the "p" we are not really incurring any additional risk. Of course, in a perfect world it would be better to set up encryption for both sending and receiving mail and then keep the "p", but at this point such a task is beyond the scope of this document. So we remove the "p" to make things easier.

Now, find this line near the bottom of the file:

```
DAEMON_OPTIONS(`Port=smtp, Addr=127.0.0.1, Name=MTA')dnl
```

This line does *not* begin with a "dnl", which means it is active. If you read it through, you will note the "Addr=127.0.0.1" text which tells Sendmail to listen for connections only on that IP address, also known as the loopback address. Since the loopback address is used to process connections to/from the same machine, by only allowing connections to the loopback address this line effectively tells Sendmail not to accept any connections from anywhere else.

By definition (since you're going through this document), you do want to allow other systems to connect to yours to send mail. To deactivate this line, you could delete it, but the safer way (in case you ever want to know what you did or temporarily lock out outside users) is to comment it out by adding a "dnl" at the beginning of the line.

Remember, previously we *removed* the "dnl" characters at the beginning of the line. In this case, we need to *add* them.

dnl DAEMON_OPTIONS(`Port=smtp, Addr=127.0.0.1, Name=MTA')dnl

Create a New sendmail.cf File

You will now create and put in place your new sendmail.cf file. Some of these operating systems require the file to be placed in /etc and others in /etc/mail, so for safety's sake place the file in both locations (it won't hurt you). Issue these commands (you *did* create a backup before changing anything, right?):

```
cp /etc/mail/sendamil.cf /etc/mail/sendmail.orig
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

6.1.2. Setting Up Queue E-mail Aliases

Setting up queues will be a several step process, which entails setting up e-mail aliases for the group specific queues and creating the queues In **/etc/aliases** the Request Tracker V3 queue e-mail entries should look something like this:

```
architecture:         "|/opt/rt3/bin/rt-mailgate --queue A_General --action correspond \
 --url http://helpdesk1.digitalglobe.com/"
architecture-comment: "|/opt/rt3/bin/rt-mailgate --queue A_General --action comment \
--url http://helpdesk1.digitalglobe.com/"
```

Once the changes are made to /etc/aliases, than issue the following command:

```
newaliases
```

6.1.3. How to Link Ticket to Asset Scrip

This scrip will link a ticket to an asset when the ticket custom field 'server_name' is filled in with a server name. This may be useful in conjunction with ExtractCustomFieldValues

```
Description: Link ticket to asset
Condition: User Defined
Action: User Defined
Template: Global template: Blank
Stage: TransactionCreate
```

Custom condition:

```
my $trans = $self->TransactionObj->Type;
my $new_value = $self->TransactionObj->NewValue;
my $cf_id = $self->TransactionObj->Field;

if ($trans ne 'CustomField') { return 0; }
if (! $new_value) { return 0; }
my $cf = new RT::CustomField($RT::SystemUser);
my ($id,$msg) = $cf->Load($cf_id);

if (!$id) {
   $RT::Logger->crit("Could not load CF: $msg");
   return 0;
}
if ($cf->Name ne 'server_name') { return 0; }

1;
```

Custom action preparation code:

```
1;
```

Custom action cleanup code:

```
my $new_value = $self->TransactionObj->NewValue;
my $asset = RTx::AssetTracker::Asset->new($self->CurrentUser);
my ($id,$msg) = $asset->Load($new_value);
if (! $id) {
   $RT::Logger->crit("Could not load asset $new_value: $msg");
   return 0;
}
($id,$msg) = $self->TicketObj->AddLink(Type => 'RefersTo', Target => $asset-
>URI);
if (! $id) {
   $RT::Logger->crit("Could not AddLink: $msg");
   return 0;
}
1;
```

6.1.4. Sample /etc/init.d/mysqld

```
#!/bin/bash
#
# mysqld       This shell script takes care of starting and stopping
#              the MySQL subsystem (mysqld).
#
# chkconfig: - 64 36
# description: MySQL database server.
# processname: mysqld
# config: /usr/local/mysql/etc/my.cnf
# pidfile: /usr/local/mysql/var/mysqld.pid

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network


prog="MySQL"

# extract value of a MySQL option from /usr/local/mysql/etc/my.cnf
# Usage: get_mysql_option FILE VARNAME DEFAULT
# result is returned in $result
# Ugly as this is, it knows nothing of option file sections ...
get_mysql_option(){
        result=`sed -n "s/^[ \t]*$2[ \t]*=[ \t]*//p" "$1" 2>/dev/null | tail -n 1`
        if [ -z "$result" ]; then
            # not found, use default
            result="$3"
        else
            # found, still have to deal with quoting and end-of-line comments
            dequoted=`echo "$result" | sed "s/^'\([^']*\)'.*$/\1/"`
            if [ x"$dequoted" != x"$result" ]; then
                result="$dequoted"
            else
                dequoted=`echo "$result" | sed 's/^"\([^"]*\)".*$/\1/'`
                if [ x"$dequoted" != x"$result" ]; then
                    result="$dequoted"
                else
                    result=`echo "$result" | sed 's/^\([^ \t#]*\).*$/\1/'`
                fi
            fi
        fi
}

get_mysql_option /usr/local/mysql/etc/my.cnf datadir "/usr/local/mysql/var"
datadir="$result"
get_mysql_option /usr/local/mysql/etc/my.cnf socket "$datadir/mysql.sock"
socketfile="$result"
get_mysql_option /usr/local/mysql/etc/my.cnf err-log "/usr/local/mysql/log/mysqld.log"
errlogfile="$result"
get_mysql_option /usr/local/mysql/etc/my.cnf pid-file "/usr/local/mysql/var/mysqld.pid"
mypidfile="$result"

start(){
        touch "$errlogfile"
        chown mysql:mysql "$errlogfile"
        chmod 0640 "$errlogfile"
        [ -x /sbin/restorecon ] && /sbin/restorecon "$errlogfile"
        if [ ! -d "$datadir/mysql" ] ; then
            action $"Initializing MySQL database: " /usr/local/mysql/bin/mysql_install_db
            ret=$?
            chown -R mysql:mysql "$datadir"
            if [ $ret -ne 0 ] ; then
                return $ret
            fi
        fi
        chown -R mysql:mysql "$datadir"
        chmod 0755 "$datadir"
        # The reason for explicitly specifying --pid-file is that there may
        # be no such entry in my.cnf, and the default behavior will be to not
        # create it at all...
        /usr/local/mysql/bin/mysqld_safe  --defaults-file=/usr/local/mysql/etc/my.cnf
--pid-file="$mypidfile" >/dev/null 2>&1 &
        ret=$?
        # Spin for a maximum of N seconds waiting for the server to come up.
        # Rather than assuming we know a valid username, accept an "access
        # denied" response as meaning the server is functioning.
        if [ $ret -eq 0 ]; then
            STARTTIMEOUT=10
            while [ $STARTTIMEOUT -gt 0 ]; do
               RESPONSE=`/usr/local/mysql/bin/mysqladmin -uUNKNOWN_MYSQL_USER ping 2>&1`
&& break
               echo "$RESPONSE" | grep -q "Access denied for user" && break
               sleep 1
               let STARTTIMEOUT=${STARTTIMEOUT}-1
            done
            if [ $STARTTIMEOUT -eq 0 ]; then
                    echo "Timeout error occurred trying to start MySQL Daemon."
                    action $"Starting $prog: " /bin/false
            else
                    action $"Starting $prog: " /bin/true
            fi
        else
            action $"Starting $prog: " /bin/false
        fi
        [ $ret -eq 0 ] && touch /usr/local/var/mysqld
        return $ret
}

stop(){
        /bin/kill `cat "$mypidfile"  2>/dev/null ` >/dev/null 2>&1
        ret=$?
        if [ $ret -eq 0 ]; then
            sleep 2
            rm -f /usr/local/var/mysqld
            rm -f "$socketfile"
            action $"Stopping $prog: " /bin/true
        else
            action $"Stopping $prog: " /bin/false
        fi
        return $ret
}

restart(){
    stop
    start
}

condrestart(){
    [ -e /usr/local/var/mysqld ] && restart || :
}

# See how we were called.
case "$1" in
  start)
    start
    ;;
  stop)
    stop
    ;;
  status)
    status mysqld
    ;;
  restart)
    restart
    ;;
  condrestart)
```

```
    condrestart
    ;;
  *)
    echo $"Usage: $0 {start|stop|status|condrestart|restart}"
    exit 1
esac

exit $?
```

6.1.5. Sample /usr/local/mysql/bin/safe_mysqld

```
#!/bin/sh
# Copyright Abandoned 1996 TCX DataKonsult AB & Monty Program KB & Detron HB
# This file is public domain and comes with NO WARRANTY of any kind
#
# scripts to start the MySQL daemon and restart it if it dies unexpectedly
#
# This should be executed in the MySQL base directory if you are using a
# binary installation that has other paths than you are using.
#
# mysql.server works by first doing a cd to the base directory and from there
# executing safe_mysqld

trap '' 1 2 3 15                      # we shouldn't let anyone kill us

umask 007

defaults=
case "$1" in
    --no-defaults|--defaults-file=*|--defaults-extra-file=*)
      defaults="$1"; shift
      ;;
esac

parse_arguments() {
  # We only need to pass arguments through to the server if we don't
  # handle them here.  So, we collect unrecognized options (passed on
  # the command line) into the args variable.
  pick_args=$1; shift

  for arg do
    case "$arg" in
      # these get passed explicitly to mysqld
      --basedir=*) MY_BASEDIR_VERSION=`echo "$arg" | sed -e "s;--[^=]*=;;"` ;;
      --datadir=*) DATADIR=`echo "$arg" | sed -e "s;--[^=]*=;;"` ;;
      --pid-file=*) pid_file=`echo "$arg" | sed -e "s;--[^=]*=;;"` ;;
      --user=*)
        if test $SET_USER -eq 0
        then
          user=`echo "$arg" | sed -e "s;--[^=]*=;;"` ; SET_USER=1
        fi
        ;;
      # these two might have been set in a [safe_mysqld] section of my.cnf
      # they get passed via environment variables to safe_mysqld
      --socket=*)  MYSQL_UNIX_PORT=`echo "$arg" | sed -e "s;--[^=]*=;;"` ;;
      --port=*)    MYSQL_TCP_PORT=`echo "$arg" | sed -e "s;--[^=]*=;;"` ;;

      # safe_mysqld-specific options
      --ledir=*)   ledir=`echo "$arg" | sed -e "s;--[^=]*=;;"` ;;
      --err-log=*) err_log=`echo "$arg" | sed -e "s;--[^=]*=;;"` ;;
      # QQ The --open-files should be removed
      --open-files=*) open_files=`echo "$arg" | sed -e "s;--[^=]*=;;"` ;;
      --open-files-limit=*) open_files=`echo "$arg" | sed -e "s;--[^=]*=;;"` ;;
      --core-file-size=*) core_file_size=`echo "$arg" | sed -e "s;--[^=]*=;;"` ;;
      --timezone=*) TZ=`echo "$arg" | sed -e "s;--[^=]*=;;"` ; export TZ; ;;
      --mysqld=*)   MYSQLD=`echo "$arg" | sed -e "s;--[^=]*=;;"` ;;
      --mysqld-version=*)
        tmp=`echo "$arg" | sed -e "s;--[^=]*=;;"`
        if test -n "$tmp"
        then
          MYSQLD="mysqld-$tmp"
        else
          MYSQLD="mysqld"
        fi
        ;;
      *)
        if test $pick_args -eq 1
        then
          # This sed command makes sure that any special chars are quoted,
          # so the arg gets passed exactly to the server.
          args="$args "`echo "$arg" | sed -e 's,\([^a-zA-Z0-9_.-]\),\\\\\1,g'`
        fi
        ;;
    esac
  done
}

MY_PWD=`pwd`
# Check if we are starting this relative (for the binary release)
if test -d $MY_PWD/data/mysql -a -f ./share/mysql/english/errmsg.sys -a \
 -x ./bin/mysqld
then
  MY_BASEDIR_VERSION=$MY_PWD          # Where bin, share and data are
  ledir=$MY_BASEDIR_VERSION/bin               # Where mysqld is
  DATADIR=$MY_BASEDIR_VERSION/data
  if test -z "$defaults"
  then
    defaults="--defaults-extra-file=$MY_BASEDIR_VERSION/data/my.cnf"
  fi
# Check if this is a 'moved install directory'
elif test -f ./var/mysql/db.frm -a -f ./share/mysql/english/errmsg.sys -a \
 -x ./libexec/mysqld
then
  MY_BASEDIR_VERSION=$MY_PWD          # Where libexec, share and var are
  ledir=$MY_BASEDIR_VERSION/libexec   # Where mysqld is
  DATADIR=$MY_BASEDIR_VERSION/var
else
  MY_BASEDIR_VERSION=/usr/local/mysql
  DATADIR=/usr/local/mysql/var
  ledir=/usr/local/mysql/libexec
fi

MYSQL_UNIX_PORT=${MYSQL_UNIX_PORT:-/tmp/mysql.sock}
MYSQL_TCP_PORT=${MYSQL_TCP_PORT:-3306}
user=mysql

# Use the mysqld-max binary by default if the user doesn't specify a binary
if test -x $ledir/mysqld-max
then
  MYSQLD=mysqld-max
else
  MYSQLD=mysqld
fi

# these rely on $DATADIR by default, so we'll set them later on
pid_file=
err_log=
SET_USER=0

# Get first arguments from the my.cnf file, groups [mysqld] and [safe_mysqld]
# and then merge with the command line arguments
if test -x ./bin/my_print_defaults
then
  print_defaults="./bin/my_print_defaults"
elif test -x /usr/bin/my_print_defaults
then
  print_defaults="/usr/bin/my_print_defaults"
elif test -x /usr/bin/mysql_print_defaults
then
  print_defaults="/usr/bin/mysql_print_defaults"
```

```
else
  print_defaults="/usr/local/mysql/bin/my_print_defaults"
fi

args=
parse_arguments 0 `$print_defaults $defaults mysqld server safe_mysqld`
parse_arguments 1 "$@"

if test ! -x $ledir/$MYSQLD
then
  echo "The file $ledir/$MYSQLD doesn't exist or is not executable"
  echo "Please do a cd to the mysql installation directory and restart"
  echo "this script from there as follows:"
  echo "./bin/safe_mysqld".
  exit 1
fi

if test -z "$pid_file"
then
  pid_file=$DATADIR/`/bin/hostname`.pid
else
  case "$pid_file" in
    /* ) ;;
    * )  pid_file="$DATADIR/$pid_file" ;;
  esac
fi
test -z "$err_log"  && err_log=$DATADIR/`/bin/hostname`.err

export MYSQL_UNIX_PORT
export MYSQL_TCP_PORT


NOHUP_NICENESS="nohup"
if test -w /
then
  NOHUP_NICENESS=`nohup nice 2>&1`
  if test $? -eq 0 && test x"$NOHUP_NICENESS" != x0 && nice --1 echo foo > /dev/null 2>&1
  then
    if test $NOHUP_NICENESS -gt 0
    then
      NOHUP_NICENESS="nice --$NOHUP_NICENESS nohup"
    else
      NOHUP_NICENESS="nice -$NOHUP_NICENESS nohup"
    fi
  else
    NOHUP_NICENESS="nohup"
  fi
fi

USER_OPTION=""
if test -w /
then
  if test "$user" != "root" -o $SET_USER = 1
  then
    USER_OPTION="--user=$user"
  fi
  # If we are root, change the err log to the right user.
  touch $err_log; chown $user $err_log
  if test -n "$open_files"
  then
    ulimit -n $open_files
  fi
  if test -n "$core_file_size"
  then
    ulimit -c $core_file_size
  fi
fi

#
# If there exists an old pid file, check if the daemon is already running
# Note: The switches to 'ps' may depend on your operating system
```

```
if test -f $pid_file
then
  PID=`cat $pid_file`
  if /usr/bin/kill -0 $PID > /dev/null 2> /dev/null
  then
    if /bin/ps p $PID | grep mysqld > /dev/null
    then    # The pid contains a mysqld process
      echo "A mysqld process already exists"
      echo "A mysqld process already exists at " `date` >> $err_log
      exit 1
    fi
  fi
  rm -f $pid_file
  if test -f $pid_file
  then
    echo "Fatal error: Can't remove the pid file: $pid_file"
    echo "Fatal error: Can't remove the pid file: $pid_file at " `date` >> $err_log
    echo "Please remove it manually and start $0 again"
    echo "mysqld daemon not started"
    exit 1
  fi
fi

#
# Uncomment the following lines if you want all tables to be automaticly
# checked and repaired at start
#
# echo "Checking tables in $DATADIR"
# $MY_BASEDIR_VERSION/bin/myisamchk --silent --force --fast --medium-check -O
key_buffer=64M -O sort_buffer=64M $DATADIR/*/*.MYI
# $MY_BASEDIR_VERSION/bin/isamchk --silent --force -O sort_buffer=64M $DATADIR/*/*.ISM

echo "Starting $MYSQLD daemon with databases from $DATADIR"

# Does this work on all systems?
#if type ulimit | grep "shell builtin" > /dev/null
#then
#  ulimit -n 256 > /dev/null 2>&1            # Fix for BSD and FreeBSD systems
#fi

echo "`date +'%y%m%d %H:%M:%S  mysqld started'`" >> $err_log
while true
do
  rm -f $MYSQL_UNIX_PORT $pid_file   # Some extra safety
  if test -z "$args"
  then
    $NOHUP_NICENESS $ledir/$MYSQLD $defaults --basedir=$MY_BASEDIR_VERSION
--datadir=$DATADIR $USER_OPTION --pid-file=$pid_file --skip-locking >> $err_log 2>&1
  else
    eval "$NOHUP_NICENESS $ledir/$MYSQLD $defaults --basedir=$MY_BASEDIR_VERSION
--datadir=$DATADIR $USER_OPTION --pid-file=$pid_file --skip-locking $args >> $err_log
2>&1"
  fi
  if test ! -f $pid_file             # This is removed if normal shutdown
  then
    break
  fi

  if true
  then
    # Test if one process was hanging.
    # This is only a fix for Linux (running as base 3 mysqld processes)
    # but should work for the rest of the servers.
    # The only thing is ps x => redhat 5 gives warnings when using ps -x.
    # kill -9 is used or the process won't react on the kill.
    numofproces=`ps xa | grep -v "grep" | grep -c $ledir/$MYSQLD`
    echo -e "\nNumber of processes running now: $numofproces" | tee -a $err_log
    I=1
    while test "$I" -le "$numofproces"
    do
      PROC=`ps xa | grep $ledir/$MYSQLD | grep -v "grep" | sed -n '$p'`
```

```
        for T in $PROC
        do
          break
        done
        #    echo "TEST $I - $T **"
        if kill -9 $T
        then
          echo "$MYSQLD process hanging, pid $T - killed" | tee -a $err_log
        else
          break
        fi
        I=`expr $I + 1`
    done
  fi

  echo "`date +'%y%m%d %H:%M:%S'`  mysqld restarted" | tee -a $err_log
done

echo "`date +'%y%m%d %H:%M:%S'`  mysqld ended" | tee -a $err_log
echo "" | tee -a $err_log
```

6.1.6. All Privileges Shell Script

The following shell scripts grants all permissions to the Request Tracker V3 **root** account:

```
#!/bin/bash

perl -I/opt/rt3/lib -MRT -e'
        RT::LoadConfig(); RT::Init();
        my $u = RT::User->new($RT::SystemUser);
        $u->Load("root");
        print $u->SetPassword("password");
'

perl -I/opt/rt3/lib -MRT -e'
        RT::LoadConfig(); RT::Init();
        my $u = RT::User->new($RT::SystemUser);
        $u->Load("root");
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "AdminAllPersonalGroups"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "AdminCustomField"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "AdminGroup"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "AdminGroupMembership"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "AdminOwnPersonalGroups"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "AdminQueue"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "AdminUsers"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "AssignCustomFields"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "CommentOnTicket"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "CreateSavedSearch"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "CreateTicket"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "DelegateRights"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "DeleteTicket"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "EditSavedSearches"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "LoadSavedSearch"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "ModifyACL"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "ModifyCustomField"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "ModifyOwnMembership"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "ModifyQueueWatchers"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "ModifyScrips"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "ModifyTemplate"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "ModifyTicket"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "OwnTicket"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "ReplyToTicket"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "SeeCustomField"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "SeeGroup"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "SeeQueue"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "ShowACL"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "ShowConfigTab"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "ShowOutgoingEmail"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "ShowSavedSearches"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "ShowScrips"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "ShowTemplate"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "ShowTicket"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "ShowTicketComments"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "StealTicket"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "SuperUser"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "TakeTicket"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "Watch"
        );
        print $u->PrincipalObj->GrantRight(
                Object => $RT::System,
                Right => "WatchAsAdminCc"
        );
'
```

6. Performance Tuning

6.1. Introduction

Properly configured, Request Tracker can run quickly on minimal hardware. This section of this document is an attempt to guide the reader through small parameter changes for the operating system, database configuration and the Request Tracker installation which can make a big impact on performance.

NOTICE: This section was provided by the Best Practical Web site at http://www.bestpractical.com.

Version Support

Comments provided here are for Request Tracker versions 3.0.10 or better.

Overview

The major components of any Request Tracker system are:

- Hardware
- Operating system
- A properly installed database application
- The Request Tracker package, release 3.0.10 or better
- All the perl CPAN modules required by that release of Request Tracker
- A properly installed web server
- Practical localized considerations for your business

Improper installation of any of the above items will lead to performance issues. However, continued focus on any one of the above items will not always lead to improved performance. It's a combination of changes across the board that will help one obtain a well performing Request Tracker system up and running.

How fast should one expect Request Tracker to run?

Looking up a ticket should take less than a second regardless of the number of tickets in the database and assuming a reasonable length of comments and attachments. The limitation should be the speed at which the server and browser can render the HTML in the page.

Myths

FastCGI is better than **mod_perl**, **mod_perl** is better than **FastCGI**, either method of executing Request Tracker can result in good response time. The traffic in the **rt-users** mailing list has not been definitive on this point. If one have Request Tracker running, even if it’s slow, under either **FastCGI** or **mod_perl**, just leave it alone. Spending time in this area will probably not result in tremendous improvements.

NOTE: Under **Linux** running **Apache2,** the **mason_handler.fcgi** process tends to grow in size. The larger it gets, the slower Request Tracker becomes; this situation can be corrected with a regular restart of apache.

MySQL is better than **Postgres**, **Postgres** is better than **MySQL**. - Improperly configured, both databases will perform poorly. Properly configured, both will run very quickly.

6.2. Performance Enhancements by Component

6.2.1. Computer Hardware

In order to run Request Tracker, one will need a fairly good sized computer system to run Request Tracker in a reasonable fashion. Consider that what is being installed is a Database backed customer and problem management system which is going to use to help run a computing enterprise. That said, as of September 2004, $1,000 spent on a 1U server with a 2Ghz single Pentium CPU with 1GB RAM and a 7200 RPM IDE disk drive is a good platform on which to run Request Tracker. SCSI interfaces will give even better performance than IDE or SATA as they can handle parallel drive accesses better, but, it may not be worth the extra dollars. If one going to spend extra dollars in a particular area of a server, spend those dollars on RAM. Then think about faster CPU and faster disk.

6.2.2. Disk I/O

The major bottleneck in any platform chosen would be disk I/O.

So let's test that first.

On a **Linux** system one can determine the disk I/O performance by using the **hdparm**. The following is an example from a modest machine (400 MHz CPU) with an IBM 10,000 RPM SCSI drive.

```
hdparm -t /dev/sdc1
```

/dev/sdc1:

```
Timing buffered disk reads:  64 MB in  1.92 seconds = 33.33 MB/sec
```

If one is using any kind of modern hardware, then expect **at least** this level of performance. If **hdparm** returns a slower I/O rate, consider getting a better hard drive. As of September 2004 a Western Digital Caviar SE, 7200rpm, EIDE, 80GB with an 8mb cache can be purchased for around $70 US. There is no reason to skimp on disk performance.

6.2.3. RAM

Request Tracker requires running a database server. Databases perform best when their indices and frequently used data structures are memory resident. If one installed a stock operating system and stock database servers, regardless of whether it's **MySQL** or **Postgres**, one **must** make some adjustments in order to get any kind of reasonable performance.

Default UNIX/Linux installations are not "database server" installations.. Fortunately the changes to be made are simple.

The basic idea, regardless of Operating System, is that one wants to allow the databases to use as much RAM as they are comfortable using.

Here are instructions for a *RedHat Linux Fedora Core 1* installation for adjusting shared memory:

1. Edit **/etc/sysctl.conf** and add these lines:

```
kernel.shmmax=536870912
kernel.shmmni=4096
kernel.shmall=2097152
```

2. Execute the command **"/sbin/sysctl -p**" to make these changes take effect

For postgresql, do this:

1. Edit **/var/lib/pgsql/data/postgresql.conf** and change these lines:

```
shared_buffers = 6004
sort_mem = 8096
```

2. Stop and start postgresql

For **MySQL**, please look at the ***my-large.cnf*** configuration file example that is provided with the installation. It's usually in **/usr/share/doc/mysql-server*** directory. The key settings are:

```
set-variable = innodb_mirrored_log_groups=1
set-variable = innodb_log_files_in_group=3
set-variable = innodb_log_file_size=5M
set-variable = innodb_log_buffer_size=8M
innodb_flush_log_at_trx_commit=1
innodb_log_archive=0
set-variable = innodb_buffer_pool_size=16M
set-variable = innodb_additional_mem_pool_size=2M
set-variable = innodb_file_io_threads=4
set-variable = innodb_lock_wait_timeout=50
set-variable = sort_buffer=2M
```

NOTE: One really should use InnoDB tables. InnoDB is required after the Request Tracker 3.2.x release and makes lots of sense, as it enables **MySQL** to perform nearly ACID compliant transactions
Pease refer to the following URL about some potential pitfalls: http://sql-info.de/mysql/referential-integrity.html, though these are probably not directly relevant to Requst Tracker.

Stop and restart **MySQL** when the changes have been made.

WARNING: One must delete **/var/lib/mysql/ib_logfileX (/usr/local/mysql/var/ib_logfileX**), if the log file size is changed, as **MySQL** will not start up. Be careful and do not delete **ibdata1**, as this is the InnoDB database.

Memory, like disk, is also quite inexpensive. As of September 2004, 1GB of PC2700 DDR is $89 US.

NOTE: Raising the **buffer_pool_size** to a value like 256MB will speed the thing up even more. For a Request Tracker setup it is suggested to use a buffer pool at about 50% of the server’s RAM size. Increasing the **log_file_size variable,** will speed up writing to the database, as does increasing the log buffer.

6.2.4. Query Optimization

For Request Tracker 3.6.0 and older, there is a database index missing which will help performance for installs with largest numbers of tickets using custom fields.

```
CREATE INDEX ObjectCustomFieldValues3 ON ObjectCustomFieldValues (ObjectId,ObjectType);
```

As the Request Tracker environment grows, ticket display will slow down without this index. If RTx::AssetTracker is installed, than the problem is more pronounced, because of extensive use of custom fields in RTx::AssetTracker installs.

Hopefully this index will be added in 3.6.1+.

6.2.5. CPU

Before one considers a CPU replacement, they should follow the procedures outlined earlier for improving I/O, memory usage and database optimization.

Request Tracker **will** run in a speedy manner on a 1GHz or better processor; CPU is not the primary bottleneck. Before purchasing a new CPU or motherboard, upgrade the disks, move to RAID 0+1 or purchase more RAM. Time and money spent on disk and memory I/O improvements will produce significant performance gains.

6.2.6. Request Tracker Software

The single largest performance improvement made in 2004 to the Request Tracter code base was the re-coding of the perl module DBIx::SearchBuilder. The latest version can be obtained by installing the latest version:

```
perl –MCPAN -e ‘install DBIx::SearchBuilder’
```

Once the new module has been installed, then do the following:

```
/usr/local/apache2/bin/apachectl stop
rm –rf /opt/rt3/var/mason_data/obj
/usr/local/apache2/bin/apachectl start
```

6.2.7. Additional MySQL Database Tuning

6.2.7.1. Introduction

After nearly two years in production, that Request Tracker mySQL database was reviewed to improve performance. This short section describes the changes made to /usr/local/mysql/etc/my.cnf to improve overall performance. The changes made include those changes described earlier in this chapter concerning InnoDB parameters.

6.2.7.2. Methodology

Review of the database performance was conducted by using a tool called mysqlard which is available for free from http://gert.sos.be/en. The tool was modified to generate graphs in “real time”, to produce legends on the graphs and general spelling corrections. The resulting tool provided various MySQL statistics and parameters settings.

The mysqlard software was installed on brutus.digitalglobe.com and was set up to monitor the database on helpdesk.digitalglobe.com. The data collection, database collection and mysqlard web page all reside on butus.digitalglobe.com. The mysqlard web page can be accessed at http://brutus.digitalglobe.com/mysqlard.

Based upon the statistics gathered, and by information produced by mysqlard, it was determined that several parameters needed to be increased. Also, it was determined that data bases caching was not active on the database. By increasing the suggested parameters, and by turning on database caching, database performance increased between 50% and 67%, in regards to record access.

6.2.7.3. my.cnf

The following is a copy of my.cnf, the MySQL configuration file. Updated parameters are in bold (these include suggested parameter changes earlier in this chapter).

```
# Example MySQL config file for medium systems.
#
# This is for a system with little memory (32M - 64M) where MySQL plays
# an important part, or systems up to 128M where MySQL is used together with
# other programs (such as a web server)
#
# You can copy this file to
# /etc/my.cnf to set global options,
# mysql-data-dir/my.cnf to set server-specific options (in this
# installation this directory is /usr/local/mysql/var) or
# ~/.my.cnf to set user-specific options.
#
# In this file, you can use all long options that a program supports.
# If you want to know which options a program supports, run the program
# with the "--help" option.

# The following options will be passed to all MySQL clients
[client]
#password       = your_password
port            = 3306
socket          = /tmp/mysql.sock

# Here follows entries for some specific programs

# The MySQL server
[mysqld]
```

```
port           = 3306
socket         = /tmp/mysql.sock
skip-locking
key_buffer = 512M
max_allowed_packet = 64M
table_cache = 128
sort_buffer_size = 2M
net_buffer_length = 8K
myisam_sort_buffer_size = 8M
connect_timeout = 86400
max_connections = 125
query_cache_size = 32M
read_buffer_size = 1M
read_rnd_buffer_size = 4M
tmp_table_size = 512M
max_heap_table_size = 512M

# Don't listen on a TCP/IP port at all. This can be a security enhancement,
# if all processes that need to connect to mysqld run on the same host.
# All interaction with mysqld must be made via Unix sockets or named pipes.
# Note that using this option without enabling named pipes on Windows
# (via the "enable-named-pipe" option) will render mysqld useless!
#
#skip-networking

# Replication Master Server (default)
# binary logging is required for replication
log-bin
# log = /dg/mysql/mysql.log

# required unique id between 1 and 2^32 - 1
# defaults to 1 if master-host is not set
# but will not function as a master if omitted
server-id      = 1

# Replication Slave (comment out master section to use this)
#
# To configure this host as a replication slave, you can choose between
# two methods :
#
# 1) Use the CHANGE MASTER TO command (fully described in our manual) -
#    the syntax is:
#
#    CHANGE MASTER TO MASTER_HOST=<host>, MASTER_PORT=<port>,
#    MASTER_USER=<user>, MASTER_PASSWORD=<password> ;
#
#    where you replace <host>, <user>, <password> by quoted strings and
#    <port> by the master's port number (3306 by default).
#
#    Example:
#
#    CHANGE MASTER TO MASTER_HOST='125.564.12.1', MASTER_PORT=3306,
#    MASTER_USER='joe', MASTER_PASSWORD='secret';
#
# OR
#
# 2) Set the variables below. However, in case you choose this method, then
#    start replication for the first time (even unsuccessfully, for example
#    if you mistyped the password in master-password and the slave fails to
#    connect), the slave will create a master.info file, and any later
#    change in this file to the variables' values below will be ignored and
#    overridden by the content of the master.info file, unless you shutdown
#    the slave server, delete master.info and restart the slaver server.
#    For that reason, you may want to leave the lines below untouched
#    (commented) and instead use CHANGE MASTER TO (see above)
#
# required unique id between 2 and 2^32 - 1
# (and different from the master)
# defaults to 2 if master-host is set
# but will not function as a slave if omitted
#server-id       = 2
```

```
#
# The replication master for this slave - required
#master-host     =   <hostname>
#
# The username the slave will use for authentication when connecting
# to the master - required
#master-user     =   <username>
#
# The password the slave will authenticate with when connecting to
# the master - required
#master-password =   <password>
#
# The port the master is listening on.
# optional - defaults to 3306
#master-port     =  <port>
#
# binary logging - not required for slaves, but recommended
#log-bin

# Point the following paths to different dedicated disks
#tmpdir        = /tmp/
#log-update    = /path-to-dedicated-directory/hostname

# Uncomment the following if you are using BDB tables
#bdb_cache_size = 4M
#bdb_max_lock = 10000

# Uncomment the following if you are using InnoDB tables
#innodb_data_home_dir = /usr/local/mysql/var/
#innodb_data_file_path = ibdata1:10M:autoextend
#innodb_log_group_home_dir = /usr/local/mysql/var/
#innodb_log_arch_dir = /usr/local/mysql/var/
# You can set .._buffer_pool_size up to 50 - 80 %
# of RAM but beware of setting memory usage too high
innodb_buffer_pool_size = 128M
innodb_additional_mem_pool_size = 16M
# Set .._log_file_size to 25 % of buffer pool size
innodb_log_file_size = 32M
innodb_log_buffer_size = 16M
innodb_flush_log_at_trx_commit = 1
innodb_lock_wait_timeout = 50
innodb_mirrored_log_groups=1
innodb_log_files_in_group=3

[mysqldump]
quick
max_allowed_packet = 64M

[mysql]
no-auto-rehash
# Remove the next comment character if you are not familiar with SQL
#safe-updates

[isamchk]
key_buffer = 20M
sort_buffer_size = 20M
read_buffer = 2M
write_buffer = 2M

[myisamchk]
key_buffer = 20M
sort_buffer_size = 20M
read_buffer = 2M
write_buffer = 2M

[mysqlhotcopy]
interactive-timeout
```