

# Request Tracker Boot Camp

Kevin Falcone - Alex Vandiver

<http://download.bestpractical.com/training/2012-PDX-OSCON/>



# Ways RT gets used

- Helpdesk
- Network operations
- Bug tracking
- Customer service
- Project management
- Backoffice workflow

# How do you use RT?

- Do you use RT?
- What version of RT do you run?
- Which database do you use?
- What do you use RT to track?
- What do you want to learn today?

# A brief history of RT

# Brief History

- 0.9 1996
- 1.0 1999
- 2.0 2001
- 3.0 2003
- 3.2 2004
- 3.4 2005
- 3.6 2006
- 3.8 2008
- 4.0 2001
- 4.2 soon (we hope)

# RT 4.0

- User interface overhaul
- Ticket lifecycles
- Mobile web interface
- Online theme editor
- GMail style history folding
- RTFM integrated as “Articles”

# RT 4.0

- Full-text search
- AJAX completion of email addresses
- Better HTML mail support
- New rights management UI
- Hundreds of performance improvements and bugfixes
- More tests

# RT 4.2 (the future)

- Never trust a software vendor who talks about unreleased software...
- The “master” branch in git will become 4.2



# Version Numbers

- 4.0.5
  - Major Series
  - Minor Series
  - Release

# Today's Demo

- Introduction to RT
- Setting up a HelpDesk
- Building a working configuration from the install
- We're assuming you have a working database, webserver, email

# Basic RT concepts

# Tickets

- The basic noun in the system
- Group together metadata and change history
- Associate a number of users, in varying roles, to the ticket

# Queues

- The basic organizational unit for RT
- A ticket is always in exactly one queue
- Generally one email address is associated with each queue
- The most common abstraction for access control

# Users

- Everyone associated with a ticket is a User
- Privileged vs Unprivileged

# History

- RT records all metadata changes and email notifications in “transactions”
- History is **immutable**
- Applies to both tickets and other objects

# Walkthrough





# Your Homepage

- Your Tickets
- Tickets you can work
- Tickets you're tracking
- Quick creation
- Reminders
- Tickets by Queue
- Dashboards
- Obsessive reloading

# Creating Tickets

- Pick a Queue
- Watchers (Requestors, Ccs, AdminCcs)
- Subject, Body, Attachments
  - like an email
- Status
- Owner

# Tickets at a Glance

- Priority
- People
  - Requestors
  - Owner
- Dates

# Working a Ticket

- Comment internally
- Assign an owner
- Reply (correspond) to the customer
- Resolve the ticket when the task is complete

# Core Features

- Ownership
- History
- Dates

# Queues

- Organizing related tickets
- Let's make a Helpdesk Queue

# Parts of a Queue

- Name and Description
  - HelpDesk
- Reply Address (help@example.com)
  - public mail in and out of RT
- Comment Address (comment@example.com)
  - internal mail in and out of RT

# Parts of a Queue for Later

- Lifecycles
- Subject Tag
- Priority



# Users

- Every email into RT is a User

# Sourcing Users

- Users can be imported from LDAP sources (including AD)
  - Created at Login
  - Bulk imported
  - preferred because it provides your Privileged and Unprivileged users and keeps data up to date.

# Basic User Info

- Username
  - non-numeric
- Email
- Real Name

# Advanced User Info

- Organization
- Addresses (buildings, office #)
- Phone Numbers

# Important Checkboxes

- Let this user access RT
  - Disable/Enable the user
- Let this user be granted rights
  - Privileged vs Unprivileged

# Privileged Users

- Generally your Staff
- People who need access to the full UI

# Unprivileged Users

- Use the SelfService interface
  - limited creation / reply forms
  - limited listing of their tickets
  - much "simpler" UI

# Organizing your Users

- Groups
  - Collect Users
  - Collect Groups



# Group Fields

- Name (HelpDesk Staff)
- Description (users who work on tickets)
- Members

# Using Groups

- Queue Watchers (Roles)
  - AdminCc
  - Cc
- Other Watchers
  - Requestors
  - Owner

# Using Roles

- Useful place to hand out rights
  - AdminCcs can See Tickets
  - Requestors can Reply to Tickets
  - Owners can Modify Tickets
- Not tied to a particular group
  - Ticket level Roles

# Rights

- OwnTicket
  - now you have more people available to own the ticket
- Owner role group
- Rights combine

# Other Queue Role Rights

- Everyone
  - CreateTicket, ReplyToTicket(?)
- Requestor
  - ShowTicket, not ShowTicketComments

# Viewed as an AdminCc

- They can take ownership of the ticket
- Reply to it
- Resolve it

# Viewed as the Requestor

- Limited view of the ticket
- Sees who is working the ticket, the replies
- Sees that it was resolved

# Requestor Information

- More About Requestor
  - Standard information
  - Lists of other tickets
  - Groups



# More About Requestor

- What about other information
  - Organization
  - Address
- MoreAboutRequestorExtraInfo

# A brief diversion

```
/opt/rt4/etc/RT_SiteConfig.pm
```

```
Set($MoreAboutRequestorExtraInfo, "Organization, Address1");
```

# Other User Information

- Product Version
  - Doesn't fit in RT's standard User information
  - Synced from elsewhere in your organization
  - Useful to see on Ticket pages for your HelpDesk staff

# Custom Fields

- Name
  - Product Version
- Type
  - Enter One Value
- Values
- Applies To
- Imported from LDAP

# More Configuration

```
/opt/rt4/etc/RT_SiteConfig.pm
```

```
Set($MoreAboutRequestorExtraInfo,  
    "Organization, Address1, '__CustomField.{Product Version}__'");
```



# Ticket Custom Fields

- Just like the User CF
- Available Globally or on Queues

# Custom Fields

- Rights
  - Similar to the Queue Rights
  - Can edit based on Role or Group
  - Can have View only rights
- Owner can Edit, AdminCc and Requestors can See

# Custom Fields

- Simple Select One
- Render Type
  - Dropdown is better than select box



# Custom Field Categories

- Never use the Category text field
  - gone in 4.2
- Categories are based on
  - Simple linked lists of values
  - More complex interactions require customization

# Custom Field History

- Custom field changes on tickets are tracked
- Part of the RT audit trail

# Searches

# TicketSQL

- Custom SQL-like language for searching tickets
- Abstracts out table joins, etc
- Custom parser, not vulnerable to SQL injection
- Can edit syntax by hand in “Advanced” search tab

# Advanced TicketSQL

- `Created > '1 hour ago'`
  - Or any date/time expression we can parse
- `Owner = '__CurrentUser__'`
  - Or Requestor, or AdminCc...
- `CF.{YourCustomFieldName} = 'Whatever'`
  - May need to limit to queue first to see queue CFs



# Saving searches

- Can be saved on users or groups
- Saving on users is a global right handed out to groups (or Privileged)
- Saving on groups is a group right handed out to groups (generally on themselves)

# Charting

- Group any set of search results
- Can also be saved like searches, based on the same rights

# Modifying the homepage

- Saved searches and charts can be added to the homepage



# Dashboards

# Group dashboards

- Similar to shared homepages
- Groups of saved searches, charts, portlets
- Sometimes bookmarked instead of homepage

# Subscribing to dashboards

- Dashboards can be scheduled to be delivered by email
- For example, daily, weekly, etc, status updates

# Groups and rights, revisited

# Nested groups

- Groups can be nested into arbitrary other groups
- Helpdesk managers is a subgroup of Helpdesk workers
- Inherits all rights of being a member of Helpdesk workers

# Ownership as locking

- A ticket can only have one owner
- Grant some ticket rights based on who is the owner of the ticket
- Multiple rights around how ownership can be transferred (OwnTicket, TakeTicket, StealTicket)

# Notifications

Scripts and Templates

# Scripts

- Notifications, business rules, and workflows
- On a some condition, trigger an action
- One common action is sending a notification using a template
- RT comes with a sane set of global defaults, but often need customization
- Global and per-queue scrips
- Allows arbitrary Perl conditions and actions



# Templates

- Notifications send email to a set of recipients
- Headers and body of email controlled by Templates
- Blank templates disable notifications
- Queue templates override global ones
- Customize RT's defaults!
- Queue specific in 4.2

# RT's subject tags

```
[oscon #1234]
```

```
Queue Subject Tag = 'helpdesk'
```

```
[helpdesk #1234]
```

# Template types

- Simple templates
  - Just variable interpolation, including:
    - `{ $TicketSubject }`
    - `{ $TicketRequestorAddresses }`
    - `{ $TicketCFSLA }`
- Perl templates
  - `Text::Template`

# Email headers

```
Subject: This is a custom header  
RT-Send-Cc: extra@example.com  
X-Giraffe: yes
```

Body of the email starts here

# Email headers

Look at that blank line: ^  
Without it, RT tries to interpret  
"Look at that blank line" as a header

# Articles

# Articles

- Knowledge base or FAQ answers
- Organized into Classes (similar to Tickets in Queues)
- Classes are applied to queues

# Content of Articles

- Need to have custom fields applied to them to be useful
- Generally at least a “Fill in one text area” but may vary by Class
- Can control what headings and content are inserted when used



# Topics

- Hierarchical organization system for articles
- Can be either global or per-queue
- Articles can be in multiple topics, similar to tags

# Searching

- Classes can declare that their articles show up in a drop-down on the reply page, for quicker access
- As a further step, the article search can be removed entirely in configuration

# Extracting

- Common answers can be simply extracted to new Articles
- Choose which parts of the correspondence end up in different parts of the Article

# Lifecycles

# Statuses

- RT comes with six default statuses
  - new, open, stalled, resolved, rejected, deleted
- Freely allows all transitions between them
  - (assuming sufficient rights)
- Lifecycles allow you to alter the set of statuses, and the directed graph of which transitions are allowed
- One Lifecycle per queue

# Status transitions

- Lifecycles also allow configurable one-click status change links in the “Actions” menu
  - “Quick delete” link for spam, for example
- Can also add new site-specific rights for status transitions

# Escalation Queues

- Second Tier Queue
- Move tickets across
- separate watchers and rights

# Links



# Link types

- Depends On/Depended On By enforce ticket resolution ordering
- Merge Into is an un-reversible merge operation
- Parent / Child for hierarchy
- Refers To / Referred To By for weak relation

# Linking other objects

- Can use any URL as a link
- “Link values to” also allows CF values to link to external systems

# Resources

# RT::OnlineDocs

- Pod in browsable form
- Integrates with your RT instance
- <http://search.cpan.org/dist/RT-OnlineDocs>

# Some light reading

- `perldoc RT::Ticket`
- `perldoc etc/RT_Config.pm`
  
- Embedding Perl in HTML with Mason
- RT Essentials (old, still relevant)

# Community

- Wiki — code examples, etc
  - <http://wiki.bestpractical.com>
- Mailing lists — discussion questions
  - <http://bestpractical.com/rt/lists.html>
  - rt-users, rt-announce
- IRC — short questions
  - #rt on irc.perl.org

# Reporting bugs

- Always discuss on rt-users first
- <http://issues.bestpractical.com/>
- [rt-bugs@bestpractical.com](mailto:rt-bugs@bestpractical.com)

# Best Practical

- <http://download.bestpractical.com/training/2012-PDX-OSCON/>



# Other Content

- Extensions
- Configuration
- More advanced Scripts
- initialdata
- Writing Extensions