Linux Trouble Ticket Software

The Help Line

If your help line serves outside users, keeping track of support requests can mean the difference between a repeat customer and a lost customer. If the line serves inside employees, an efficient response means better productivity. Fortunately, several Linux-based applications offer help for your help desk or hotline.

BY JAMES MOHR

Everyone who has ever had to call to a company because of problem with some product has had to deal with a hotline. Keeping track of the calls coming in on the hotline not only helps solve problems faster but can actually help prevent some problems from happening in the first place.

Most hotlines today use some software package to manage the calls. Although the terms *hotline* and *help desk* serve the same basic purpose, they are commonly used to refer to two different things (and sometimes the same thing). The convention is that *hotline* refers to the group of people who deal with requests from the outside, such as customers. On the other hand, *help desk* is the term used for the group of people that handle internal problems.

In both cases, you are keeping track of support requests. An incoming request is typically recorded in a "ticket." So, to keep things simple, I will refer to both types of products as "ticket systems." In this article, I'll give some background on ticket systems, and I'll review some ticket system applications that are currently available for Linux.

Ticket System Basics

Ticket systems can be very simple tools that allow you to track support requests. Or they can be more complex tools that enable you to manage problems, such as initiating work orders directly from the tickets. Differentiating between trouble tickets and work orders is very useful whether dealing with external customers or other employees.

Nothing prevents you from using the same ticketing system to manage support requests as well as trouble tickets. Mixing the two can be confusing, so it is useful to be able to assign them to different groups or to provide special fields indicating whether the request is a normal ticket or a work order. Not every product makes it easy to do that, however.

A key aspect of a ticketing system is the ability to assign a unique number (or ticket ID) to each ticket. While this feature is standard among ticket systems, the way the ticket ID is generated is not always the same, and you may not always be happy with how the ID is created.

Also, ticketing systems need a way to track who is responsible for the ticket. At the very least, the ticket system will allow you to define users to whom the tickets can be assigned. Most systems take this one step further and let you create groups of users. Here, too, the terminology is often very different but the goal is to assign a ticket to a group of people so that the next available user can take it.

Generally, ticket systems are composed of three main units: users, groups, and queues (although the terminology is not consistent). A queue is the central administrative element for processing tickets. The queue represents a "line" of tickets that need to be processed. For example, you might create separate queues for network, printer, and application problems.

Running the Tests

If you perform a search on Google and SourceForge, you can quickly find a dozen different products that fall into the category of "ticketing systems." If I were to talk about every one one of them, I would only be able to write just a couple of paragraphs. So I had to narrow down the search a bit.

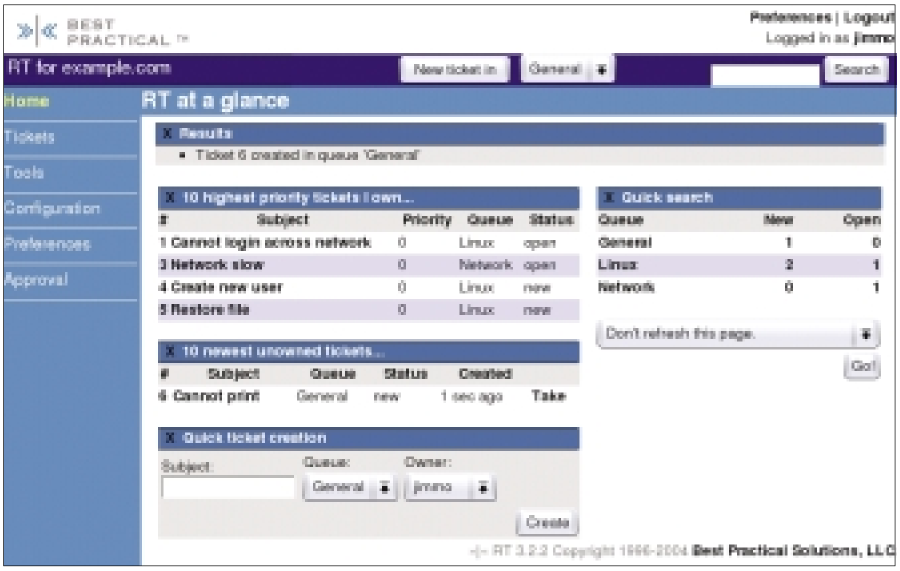
The first criterion was that the software obviously needed to run on Linux. Next, I limited it to products that are currently being developed. Here "current" is somewhat arbitrary, but I intentionally did not cover products that had not been updated recently. Some looked nice and had nice features, but who knows where they will end up after you spend the time installing them.

The final choice for which applications to include was also somewhat arbitrary. I basically looked through their feature list to see what they offered and chose the tools I thought offered the best features. I did *not* restrict myself to just non-commercial or open source products.

The interfaces I saw were exclusively web-browser based. This allows easy access by customers and support reps alike. However, if you test these products, you need to take a look at how the product reacts to your standard browser. In some cases, I found that the web interface had some glitches in different Linux browsers.

In each case, the product supported submitting tickets via email, either by having your email server send the email to a special filter (most email servers can do that) or by having the ticket system collect the messages itself using POP3 (some did both). Regardless of whether you are managing your own server or using a web hoster, one of these methods should work.

Another characteristic I looked for is the ease with which you can fit the ticket system to your needs. One part of this is the ability to restrict access to different parts of the system. For example, can you separate tickets into different groups where users only have access to tickets within their group? Or read-only access to other groups? Can you prevent users from deleting tickets? What other rights or privileges can you define? Can you configure the interface or the default responses to customers?



RT at a glance

Results

- Ticket 6 created in queue 'General'

10 highest priority tickets I own...

#	Subject	Priority	Queue	Status
1	Cannot login across network	0	Linux	open
3	Network slow	0	Network	open
4	Create new user	0	Linux	new
5	Restore file	0	Linux	new

Quick search

Queue	New	Open
General	1	0
Linux	2	1
Network	0	1

Don't refresh this page. Go!

10 newest unowned tickets...

#	Subject	Queue	Status	Created	
6	Cannot print	General	new	1 sec ago	Take

Quick ticket creation

Subject: Queue: General Owner: jimmo Create

-[- RT 3.2.2 Copyright 1996-2004 Best Practical Solutions, LLC

Figure 1: Request Tracker "At a Glance."

Fitting the system to your needs can also mean the ability to include your own types of data. That means being able to add new fields. As with user permissions, the extent to which this feature is implemented varies from product to product. Personally, I think that this is one feature that is very often underestimated.

Because I do not know your organization and your specific requirements, I cannot make a judgment as to which of these products you should use. Instead, I will give you a quick look at what is available and some of the features each product has to offer. If you really want to make an informed decision, I suggest picking out a couple of these tools and installing them on your own computer. You may also wish to visit the on-line demos that are available for some of these products.

Request Tracker

Request Tracker [1] is an Open Source ticket system from Best Practical Solutions. The installation of Request Tracker (RT) was very cumbersome. In addition to requiring Perl 5.8.3, the version I downloaded needed a long list of Perl modules, many of which I had never heard of before.

They provided a *make* file, which included features to tests/display the dependencies as well as automatically "fix" them. This takes advantage of Perl's ability to automatically download and install modules from the CPAN archive. However, this did not work completely and I had to download some modules by hand.

One annoying thing was that, if RT does not find the "required" Perl version when running *make fixdeps*, it still continues to download Perl modules for the version already on your system. According to Best Practical, this will be changed in the next release.

When you first log in to Request Tracker, you start off with "RT at a glance." This gives you a down-n-dirty overview of what's going on. Not all open tickets are available directly from the start page, but they are generally just one click away.

On this page you also have a block labeled "Quick ticket creation." As its name implies, this feature allows you to create tickets really fast. Here, all you have a choice of is the subject, the owner, and the queue. Once you click on "create," a ticket is created and you are brought back to the start page, so you can create more quick tickets or work on existing tickets.

If you take a look at Figure 1, you will see that each function has its own header with an *X* on the left end. By clicking this *X*, you "collapse" the box for that particular feature and only see the heading. Click the *X* again and the box opens up as before. This feature appears on basically all of the ticket-related pages and can be very useful if the page is "overcrowded" with information.

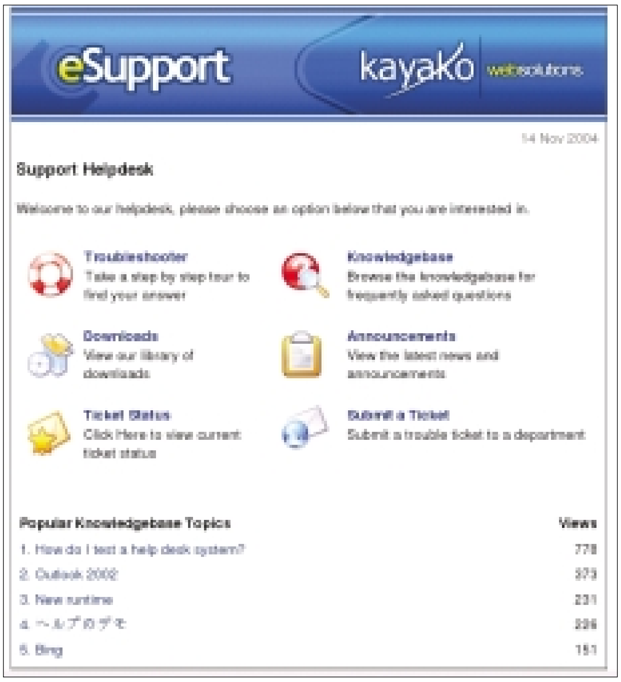
RT has a sub-class of users called "watchers." As the term implies, these are people who are watching the ticket. This can be people who are actively involved, like the requester and support rep. It can also be people who just have an interest in the ticket. For example, if the company owner opens a ticket, the support department manager might be interested in following the progress.

As with any product, tickets are processed by people with specific skills, so it make sense to send each ticket into a queue based on the skill set needed to address that issue. Thus, by default, a queue also serves as the ticket's "category." However, with RT you can create your own custom fields, such as category, classification, or anything else many commercial products provide.

As an administrative unit, groups and queues can typically be used when assigning rights. Specific people can have the right to *modify* tickets within certain queues but only *read* tickets in other queues. Certain groups can carry out specific functions. RT does a great job controlling what rights each group has and what each queue does. Since users are part of the groups, you are indirectly defining the privileges users have.

RT takes the concept of a ticket history two steps further than other products. First, in addition to a standard history, you have a history of changes related to the call – for example, when the call is assigned to a different person or the status changes. Second, you can mark non-administrative history entries as either a reply to the requester or to "private," so that the requester cannot see the reply. This feature might be useful in passing information to other support reps about the relative knowledge of the requester (or lack thereof).

Priorities can range from a low of 0 to a high of 99, and you can set the ticket to either increase or decrease in priority as the response deadline approaches. 100 priority levels might be going overboard, but you can configure the behavior of the priorities to fit your company.



14 Nov 2004

Support Helpdesk

Welcome to our helpdesk, please choose an option below that you are interested in.

Troubleshooter
Take a step by step tour to find your answer

Knowledgebase
Browse the knowledgebase for frequently asked questions

Downloads
View our library of downloads

Announcements
View the latest news and announcements

Ticket Status
Click Here to view current ticket status

Submit a Ticket
Submit a trouble ticket to a department

Popular Knowledgebase Topics	Views
1. How do I test a help desk system?	778
2. Outlook 2002	373
3. New runtime	231
4. ヘルプのデモ	206
5. Bing	151

Figure 2: The eSupport start page.

Another useful feature is the ability to create relationships between tickets. This can be simply a note to say "Hey, look at this other ticket." Or you can create actual dependencies and even parent-child relationships.

This mechanism would allow you to use RT to create a "work order" system, whereby a problem is reported and it generates a second ticket that is assigned to a second queue, which is responsible for actually carrying out the work. Or even "projects," whereby one large task is composed of several smaller tasks.

RT added an enhancement to their customer communication, which they call *scrips*. Scrips are customer notifications that react to various criteria. The classic example of this is where the support manager gets a message on his or her pager that says the company owner opened a new ticket.

Most of the products have the ability to search for specific calls. However, the RT search feature deserves special mention because of the extent to which you can define your search.

OTRS – Open Ticket Request System

Open Ticket Request System (OTRS) [2] is another Open Source ticket system. The installation for Open Ticket Request System (OTRS) is exceptionally easy if you have a Linux distribution that supports the RPM package format. Although installation required a few packages that I didn't yet have on my Suse system, these extra packages easily installed from from the Suse CDs. Once the RPM package was installed, I could use a web browser for the rest of the configuration, which included creating a database and setting up the necessary users.

The application also supports non-RPM Linux distributions and UNIX dialects, but you only have a *.tar.gz* file and have to perform all of the installation steps by hand. This includes checking for all of the necessary Perl modules, creating and configuring your database, and configuring the web server. This full installation is helpful if you want a better understanding of how the components work, but it is cumbersome if you don't.

Although privileges, users, and groups cannot be tuned as finely as with other products, you have a number of configuration options for the queues that are not available in other products, such as the option for sub-queues to organize the assignment even better, a setting for the maximum time before the call is escalated, and a setting for limiting the time calls can be locked. (Call locking was a feature few products had.)

OTRS can be configured to allow users to create their own accounts before creating tickets (called "Customer Self Registration"). A customer provides a name and email address. Should the email not arrive, the company has a record that the user contacted the hotline. Customers can also create an account by sending email.

For the most part, ORTS tickets are managed as standard email. The messages are stored in clear text on the hard disk, with the headings stored in the database. This design allows you to search for tickets more quickly. Even attachments are treated the same way as with standard email.

The appearance of the tickets within a queue is similar to an email client, with the standard *From*, *To*, and *Subject* headers. Alongside the standard headers are ticket-specific values like the state/status, priority, current queue, and so on. You can then "zoom" in the ticket to get the details.

One nifty feature is a block on the ticket page labeled "response templates," which allows you to compose freely configurable default responses, such as "Please ask Google," "RTFM," and "Read the FAQ." Things like this are extremely useful if you have a high percentage of repeat tickets or tickets that can be handled with standard answers. Different templates can be assigned to different queues if you wish to give a different response depending on the queue. Also on the ticket page is the *CustomerID*, which is a link that brings you to all open tickets from a specific customer.

OTRS does a good job of keeping track of changes in the history. Many of the administrative changes (i.e., assigning a new owner) allow you to add a note to that change that is then accessible through the history. For example, when you re-assign a call, you can add a note as to why it is being re-assigned.

OTRS does have a couple of annoyances. One annoyance is that, although you do have some custom definable fields, you are limited to only four of them. Also you are not allowed to close a call unless you are the owner. In my experience with other products, users other than the owner should be able to close a call.

eSupport

eSupport [3] is a commercial ticket system by Kayako Web Solutions. I thought the Request Tracker installation was cumbersome, but the installation of eSupport was a real head-shaker. Although there were fewer steps to go through – I didn't need to download dozens of modules, and a lot of the configuration was done through the web browser – there was *a lot* of guessing involved about where to put things and how to start the installation.

For a product that asks you to pay real money, the installation documentation is poor, which made the installation that much harder. The answer to questions I asked the company was basically "It's in the doc" or "It's on the website." When I asked where in the doc is was, they couldn't provide an answer, and I still haven't found the answer on their website.

As with other products, you have users and groups (which are called "departments"), and users only see calls that are assigned to groups of which they are members. You can move tickets from one department to another, and the fact it was moved is recorded in the history along with the name of the user. However, the details are not automatically included (for example, "moved from group A to group B").

One saving grace for eSupport is the very easy-to-use interface. This applies to not only the user interface (Figure 2), but also the administration. Although you cannot configure as much as you can on some non-commercial products, a lot of the configuration is more self-explanatory. However, as with the installation, I still had to do a lot of guessing as well as "trial and error."

Another useful aspect of eSupport is the "knowledge base." As with other products, this is much closer to a classical FAQ than to a knowledge base, in that you have a specific answer to a given question. Users can extend the knowledge base by adding comments or ratings to the entries. By default, all comments are moderated.

The knowledge base interface makes it easy to navigate, create new categories, create questions and answers, and so forth. Also, you can add HTML code to the answer, creating (for example) links to more detailed answers. Plus you can have links to "related topics" (i.e., other articles).

There is also a troubleshooter feature. Like its namesake in Windows, the troubleshooter leads you through a series of questions to help narrow down the problem. The process takes time, but if you have a lot of fairly easy-to-solve problems, the troubleshooter is a useful feature.

The product also has a couple of reporting features. You can create a report of what each user is working on. You can also generate "statistics" for a given "client." The statistics report lists open calls with information on each call, such as the subject, creation date, and so forth.

The usefulness of the reporting was diminished by the fact that the "client" is the first person to submit any call for any given email address. I successfully created multiple calls for a single email address, but with different names. The program did not say anything about this (let alone prevent me). When generating the statistics, only the first name used was displayed, so getting the statistic for a single person is not that easy.

An interesting feature is called *InstaSearch*. This feature searches the knowledge base for *possible* answers to a problem. These answers can be sent along with auto-responder emails, as well as when the user submits a ticket. Unfortunately, from both my own experience and what others have said on the eSupport forums, there may be so many items that the submit button for the ticket disappears off the screen. As a result, the person thinks that the ticket was already submitted, when, in fact, it never was.

perlDesk

The commercial perlDesk [4] system is very simple and easy to use. This simplicity also means that perlDesk does not have many of the same features that other tools have, even some of the open source tools. However, perlDesk provides a very professional look-and-feel. Whereas with some products you are obviously working with a web browser, perlDesk gives the impression of a "real" application.

As with eSupport, the lack of documentation in a commercial product was disappointing. The perlDesk installation instructions were better, but that's about where the documentation stopped. However, the support provided by the perlDesk developer (logicNow Limited) was exemplary and they definitely made an effort to ensure I got things working (in stark contrast to the other commercial product).

Users can submit calls either via email or the web interface, which also provides a self-registration mechanism called "Quick User Signup," similar to OTRS.

One really useful feature was the ability to add a ticket directly to the knowledge base (FAQ). When you click the submit button, you are brought to a new form, where you can choose the FAQ category as well as edit the text, which does not affect the text in the original ticket.

The FAQ works basically the same way as with other products, with a set of fixed answers to known problems. This is supported by the "troubleshooter" component, which also works the same way as with other products. Adding new entries to the troubleshooter is very easy.

One nice feature is the ability to do a database backup from within the product itself. If you are running the server yourself, you may not necessarily need this feature. However, it is really useful if you are using a web services provider and that is the only way you can save your data.

Another nifty feature is the *Staff Rating*. When *Staff Rating* is enabled, users can rate the responses by the support staff (on a 1-5 scale). The user can also add comments to that rating. To me, this is more useful in determining the effectiveness of your staff than simply looking at the number of tickets closed.

As simply a matter of personal preferences, I did not like the fact that the administration is completely separate from the staff. In order to administer the system, you have a completely different login. Although it helps to separate the functions, it can be an annoyance if you need to make a change in the middle of processing a ticket.

Disappointing also was perlDesk's inability to search for specific tickets. You can look for all open (unresolved) or all closed (resolved) tickets, as well as display open tickets by priority, department, and staff member. However, that is about it.

One feature of perlDesk that was not readily available in many other products was the ability to change the layout of the pages through a template editor. This is done very simply with a *textarea* in an HTML form, but you can easily configure basically every page in the system.

OSTicket

The Open Source OSTicket system [5] was definitely the easiest to install. I did not have to download any other packages or files from other sites. I didn't have to perform detailed configuration of my Web Server. I had the application up and running in less than five minutes. The only place in the installation process I would call cumbersome in the least was the fact that I had to create the database by hand. However, since this was also required for commercial products, I can easily overlook it.

The product is a slim, yet effective ticketing system with few features. Unpacked, the package takes up just over 300K(!) on your hard disk, so you obviously cannot expect the same number of features you get with a product that is 10 times the size. However, depending your your needs, the small size of the OSTicket application could be a benefit.

Note that the 300K does not include the documentation, as almost none is provided. Instead you are directed to the OSTicket forum for help. This is something the developers say will be improved in the upcoming release. Nonetheless, the interface is fairly straight forward.

Other than managing tickets, this product does little else. You can create groups and users (called representatives), and limit which category of calls the user can access, as well as which aspects of the system users can configure (although there aren't that many). But here the features slowly dwindle.

Tickets are grouped into categories that are then assigned to specific groups. Members of the group then have access to the ticket in the categories. Users are limited to a single group.

One really annoying aspect of OSTicket is that no matter where you start, you are invariably brought back to the main page. For example, when creating a new user, clicking on the *Create* button brings you to the main page and not to the new user page. This is especially annoying when you forgot to input a required field. Granted you can simply press the *Back* button in your browser, but it is more likely than not that you will want to correct the mistake by returning to the page.

One thing that really impressed me when researching this article was the fact that there are a number of web sites that offer OSTicket web hosting. This demonstrates the ease with which OSTicket can be installed and managed.

I was occasionally bothered by the limited number of features. However, if all you are interested in is an easy way of keeping track of any kind of request, OSTicket is definitely worth a look.

Conclusion

The ticket systems described in this article provide a range of useful features. The ideal system for you will depend on your own requirements. Take some time to investigate the available options, and choose the system that is right for your environment. ■

THE AUTHOR

James Mohr is responsible for the monitoring of several datacenters for a business solutions provider in Coburg, Germany. In addition to running the Linux Tutorial web site (http://www.linux-tutorial.info), James is the author of several books and dozens of articles on a wide range of topics.

INFO

[1] Request Tracker *http://www.bestpractical.com/rt/*

[2] Open Ticket Request System *http://otrs.org/*

[3] eSupport *http://www.kayako.com/?_a=products&_m=esupport.features#*

[4] perlDesk *http://www.perldesk.com/index.html*

[5] OSTicket *http://www.osticket.com/*