

The Open Ticket Request System, Free Software for the Helpdesk

Help Is On Hand



When the demand for user support exceeds a certain limit, a Trouble Ticket system becomes inevitable, if you want to stay efficient. OTRS is an Open Source system that facilitates the processing of support requests that has already proved its value for enterprises handling tens of thousands of support requests daily.

BY STEFAN WINTERMEYER

Enterprise organizations with non-trivial products and services that require some explanation are often measured by the quality of the support they offer. Support can make massive inroads on such an organization's resources: automated systems that structure the flow of support requests and responses, so-called Trouble Ticket systems, are required to prevent this from happening.

Using a Trouble Ticket system can mean a genuine competitive advantage, a fact that is often reflected in the prices asked by its manufacturer. However, this is not true of OTRS [1], a Free software package available under the GPL at a licensing charge of zero.

Reduce Waiting Lists

The GPL Open Ticket Request System (OTRS) pools incoming mail, phone calls and faxes at a central pickup address, where a team of support workers can collect them for processing. One of the major challenges is assimilating requests that arrive by various media, such as phone, email, fax, or even snail-mail, and assigning tickets to them.

A ticket contains all the data pertaining to a support request. OTRS tickets for phone calls can be assigned either manually, or semi-automatically via an interface to the phone system (LTAPI). Faxes are converted to GIF format and attached to an email message. Letters are handled the same way as faxes (you fax them to yourself). Ultimately, an OTRS ticket will always be an email message.

Version 1.1 was released on the 1st of May, and we took this opportunity to take a closer look at the software project.

Take a Number!

The Trouble Ticket Number (TTN) is the unique identifier for each ticket. When a customer sends an email message to *hotline@company.com*, she receives an OTRS response with the TTN in the subject line. If the customer replies to this message, OTRS uses the TTN to identify the ticket and can handle it accordingly.

A TTN can comprise of letters and numbers, although it makes sense to keep to numbers only. This allows the customer to use a voice line or touchtone interface to enter her number directly. The telephone exchange interface (LTAPI) can then put the customer through to the right exchange, where the person in charge should already have the ticket on her screen when the call comes in. One important thing about the TTN is the fact that it must not be modified.

OTRS provides a selection of algorithms for creating TTNs, the most common method being *\$year\$month\$day\$ticketcount\$validation* (for example, 20020323000109). If you want to prevent customers from ascertaining the number of support requests on a specific day by analyzing the TTN, you can easily write a TTN module of your own and add this to the extremely modular OTRS structure.

Web-based Client

Many commercial Trouble Ticket systems use special clients that often need

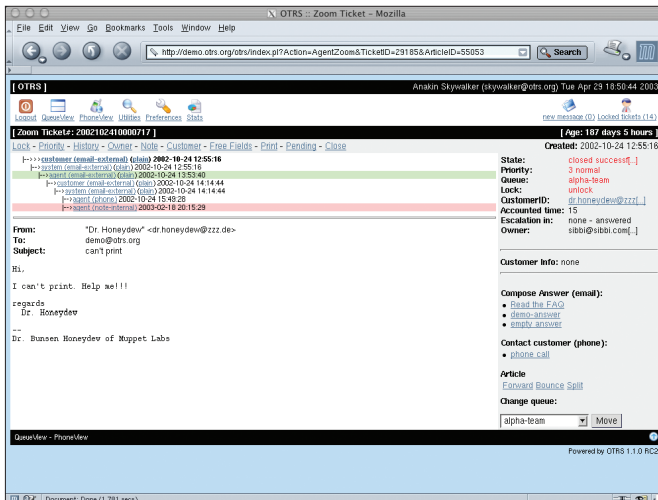


Figure 1: Queue View shows just how long some problems take

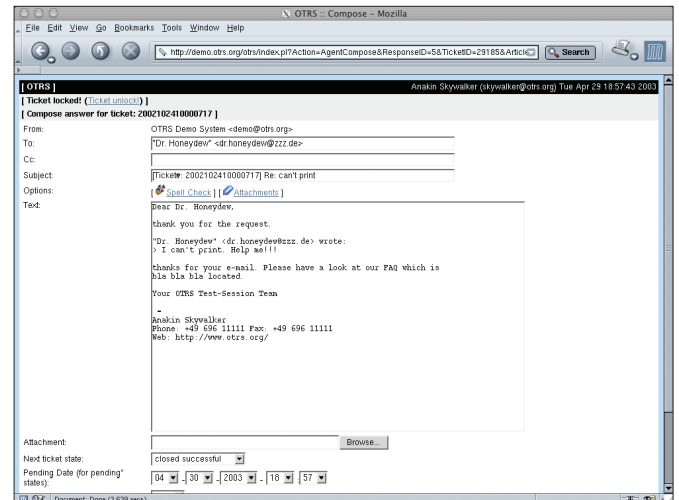


Figure 2: Standard responses can be created

Windows. With OTRS, the user (agent) needs only an account on the system and a web browser. Even Lynx and w3m produce useful results. The user can then log on and view any queues for which she has been assigned access privileges.

After installing the system, the system administrator can also perform system configuration tasks using a webclient. The client is intuitive and easily adapted to suit a user's requirements. Templates (written in the dynamic template language) are available for each individual page, and can easily be modified to reflect your corporate design. This also ensures that the agent environment is unaffected by any software updates.

Ticket Queues

Individual tickets are pooled in queues. The administrator can specify what agents have access to a queue, thus ensuring that queries addressed to accounts are exclusively handled by accounts and do not land on the technical support desk, despite being part of a single system. In the case of larger installations a dispatcher unit is often added to the RAW queue.

The RAW queue accepts any incoming customer queries, and the dispatcher distributes the mail messages to more specific queues. If a dispatcher mistakenly moves a customer query to a 2nd level support queue, although it should have landed in 3rd level support, agents in the 2nd level queue can add a comment to the ticket.

Of course, the comment is also stored in an appropriate email folder, and the

ticket is returned to the dispatcher or to 3rd level support. A sophisticated system of privileges for each queue decides who can move what ticket to or from what queue.

The Default Case

Most queries can be handled by an FAQ response. This is where OTRS shines, as you can define any number of standard responses for each queue. However, from a practical point of view, you should avoid defining more than eight responses per queue.

If you want to use a standard response to reply to an email, you can simply click on the appropriate link to open a Web based email client that already contains the customer query, the standard response and a personalized greeting.

Under ideal circumstances, you just click on "Send"; alternatively, you can edit the response. This allows many systems to handle up to 70 per cent of all responses quickly and simply. However, a system of this type is highly dependent on its maintenance state. Don't forget to update OTRS when standard responses change.

Queue Robots

The ability to employ queue handling robots is a particularly neat feature that most support staff will appreciate. Version 1.1.1 (codename Bondi Beach) provides a "GeneriC Agent" that can automatically sign off, move or comment on tickets in a queue. According to the OTRS wishlist, robots will soon be able to pipe the content of a ticket

through an external application in future, and this should allow you to apply filtering.

Thus, a RAW queue could be searched automatically for keywords and any matching tickets placed in appropriated queues. Another application would be to automatically sign off spam tickets offloaded into a spam queue by the dispatcher.

Again the developers are looking for a high degree of automation at this point. A dispatch agent can move a ticket to a spam queue with a single click, and the robot can then add an appropriate comment and sign off the ticket. Many OTRS installations also use software such as Spam Assassin to automatically move spam to a spam queue (this is done by adding a special X header that OTRS parses).

Solutions à la OTRS

You can specify an escalation interval to avoid having tickets waiting for too long in queues. When the escalation timer expires, an agent working on this queue can process only this ticket, thus ensuring that tickets do not become stale.

Depending on the field of application, it can make sense to re-process a ticket after a certain time, either to sign the ticket off or phone the customer again. Thus, the system allows you to define a follow-up interval on a ticket by ticket basis, or to use a default interval.

One of the major advantages of a closed Trouble Ticket system is the capability of precise transaction logging which allows you to adhere to service

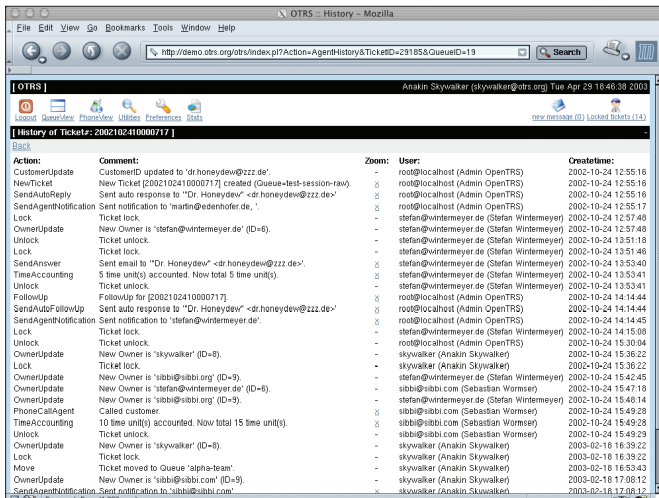


Figure 3: The history of a ticket is easily viewable by using a standard browser

level agreements entered into with customers. This allows you to clearly demonstrate what activities occurred at any specific time.

The “sick agent” scenario is a typical problem that large-scale Trouble Ticket systems suffer from.

An agent locks a ticket, thus securing exclusive access to the ticket. Now, if the agent gets sick a day later, and the ticket is still locked, other agents will be unable to respond to the ticket.

To prevent this from happening, you can set a timer on a queue by queue basis to automatically unlock any locked tickets after a certain interval, thus allowing unrestricted access by any agent.

If you intend to get the best out of OTRS, you can look forward to a marathon configuration session. The sheer variety of configuration options reminded us of the Mutt email client. You can customize more or less everything. OTRS is also really cosmopolitan, allowing you to specify a language for each agent’s menu system and a codeset for the display. The OTRS trial system at <http://www.otrs.org/> demonstrates this quite effectively using a few sample mail messages in Russian.

What Database and What Size?

At present both the MySQL and Postgres databases are supported. If you have been following the OTRS mailing list, you will have noticed that there are supporters in both camps. The largest

installation, run by a major British ISP, uses MySQL, and the system handles 20,000 tickets a day. The installation with the largest number of agents is run by a major German bank. Up to 2,000 service staff are available at any time, and all of them use OTRS. Once again, MySQL is the database used here.

Of course, most installations are a lot smaller (handling between 10 and 500 email messages per day), but it is always nice to know that the system is scalable and can handle far more demand than you need.

For statistics fans, OTRS provides a whole bunch of useful report formats. However, union representatives may not entirely approve of this kind of report, so you might like check your legal status first.

Installation Pains

Installing the system is by no means trivial. If you are short on time, don’t even think about it! OTRS requires a web-server (typically Apache), a database, and Perl (version 5.5 or later). Installing OTRS on SuSE Linux is the easiest way to go at present. The development team provides a downloadable RPM for this purpose. This should allow you to install the software within a few minutes.

Other Linux and UNIX variants will have to make do with the *tar.gz* file, and install the software manually. You can expect to have to install at least half a dozen Perl modules via CPAN first. Having done so, you can go on to modify *httpd.conf*, restart your Apache

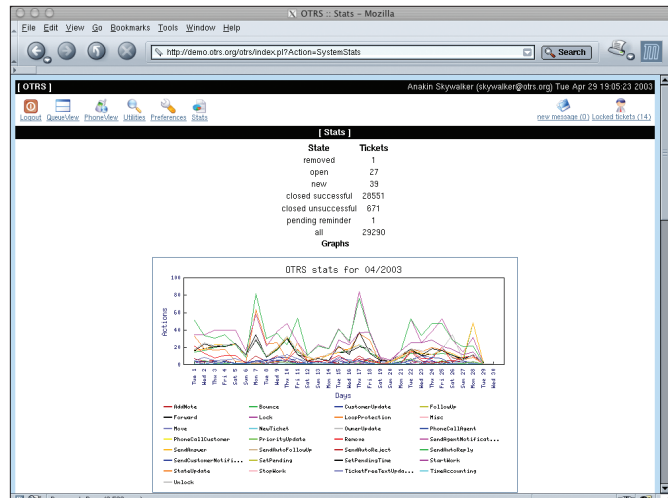


Figure 4: OTRS provides clear cut statistics that quickly reveal staff who like to take things easy

(important!) and install the database. However, the documentation area on the OTRS homepage does provide a useful installation guide.

Don’t worry about the low version numbers, by the way. This seems to reflect the developer’s slightly twisted sense of humor. Versions referred to as beta in the OTRS tree would easily make the grade as final releases for most software developers. It’s the same humor which creates the codenames for the releases: 1.0.1 was Sunset Beach, located at the North shore of Oahu, Hawaii, 1.1.1 Bondi Beach, located at Sydney, Australia.

If you are not sure whether OTRS is the right system for a task in hand, you can experiment with the trial system on the project homepage. Of course, the project homepage also provides more details.

INFO

[1] OTRS homepage: <http://www.otrs.org>

THE AUTHOR

Stefan Wintermeyer works as a Vice President for TechSpan Enterprise in Bad Homburg, Germany. He used to work as a project manager for Lufthansa in Frankfurt, Germany. Before that he was Vice President at SuSE Linux AG in Nuremberg, Germany. He dedicates much of his spare time to Open Source projects.

