

OTRS - Manual

Martin Edenhofer

OTRS core team

martin@otrs.org

Stefan Wintermeyer

OTRS core team

stefan@otrs.org

Sebastian Wormser

OTRS core team

sibbi@sibbi.com

OTRS - Manual

by Martin Edenhofer, Stefan Wintermeyer, and Sebastian Wormser

Three Tables Beach Edition

Published 2003

Copyright © 2003 by Martin Edenhofer and Stefan Wintermeyer

The copyright holders make no representation about the suitability of this document for any purpose. It is provided “as is” without expressed or implied warranty.

Dedication

This manual is dedicated to the nice folks of *Cafe Lucas*(<http://www.cafe-lucas.de/>) and *Enchilada*(<http://www.enchilada.de/>) (two restaurants in Nuernberg). Thanks for the happy hour! Today we hang out mostly in Frankfurt but we still remember the good times in Nuernberg.

Table of Contents

Foreword.....	i
1. Install the biest.....	1
1.1. Installing the rpm on a SuSE Linux (the quick and easy way).....	1
1.2. Using the tar.gz file to install OTRS on any Linux/Unix platform.....	4
1.2.1. Install.....	4
1.2.2. Database.....	6
1.2.3. Webserver.....	8
1.3. Install OTRS on Win32.....	9
1.3.1. Installations.....	10
1.3.1.1. Install PERL.....	10
1.3.1.1.1. Install Additional PERL Packages.....	11
1.3.1.2. Choosing And Installing The Webserver.....	11
1.3.1.2.1. Installing Apache.....	11
1.3.1.2.2. Install Sambar Server.....	11
1.3.1.3. Install MySQL.....	12
1.3.1.4. Install OTRS.....	12
1.3.2. Configuration.....	12
1.3.2.1. Configuring Perl.....	12
1.3.2.2. Configuring The Webserver.....	13
1.3.2.2.1. Configuring Apache.....	13
1.3.2.2.2. Configuring Sambar.....	14
1.3.2.2.2.1. Securing /otrs/bin.....	15
1.3.2.3. Configuring MySQL.....	16
1.3.2.4. Configuring OTRS.....	16
1.3.2.4.1. CONFIG.PM.....	16
1.3.3. The First Start.....	17
1.3.3.1. System Email Addresses & POP3.....	17
1.3.3.2. Setup CRON Job(s).....	17
1.3.4. Done!.....	18
2. Basics about a Trouble Ticket System.....	19
2.1. A simple example for a small Trouble Ticket System.....	19
2.2. What is a trouble ticket in the OTRS?.....	19
2.3. What is a ticket queue?.....	19
3. First steps.....	20
3.1. Login as root and create a new account.....	20
3.2. Login with user privilege.....	21
4. Ticket topics.....	23
4.1. Answer ticket via email.....	23
4.2. Answer ticket via phone.....	23
4.3. Zoom ticket.....	24
4.4. History of ticket.....	24
4.5. Add note to ticket.....	24
4.6. Close ticket.....	25
4.7. Forward ticket.....	25
4.8. Ticket priority.....	26
4.9. Ticket print view.....	26
5. System-Email-Notifications.....	28
5.1. Agent-Notifications.....	28
5.1.1. new ticket.....	28
5.1.2. ticket follow up.....	28
5.1.3. new ticket in custom queue.....	28
5.1.4. ticket lock timeout.....	28
5.2. Customer-Notifications (min 1.1).....	28
5.2.1. ticket state.....	28
5.2.2. ticket owner.....	28
5.2.3. ticket queue.....	28

6. Auto-Response.....	29
6.1. Reply	29
6.2. Follow up	29
6.3. Reject	29
6.4. Closed -> new Ticket	29
7. Receiving emails.....	31
7.1. Via POP3 account	31
7.2. Via command line program and e. g. procmail.....	31
7.2.1. Queue dispatching with OTRS	32
7.2.2. Queue dispatching with procmail (for more complex dispatching).....	32
7.2.3. Example with procmail and a webform.....	33
7.2.4. Fetching emails via POP3 or IMAP and fetchmail.....	34
8. Sending emails.....	35
8.1. Via Sendmail (default)	35
8.2. Via SMTP relay/smarthost (min. OTRS 1.1)	35
9. User.....	36
9.1. How it works	36
9.1.1. Admin-Interface.....	36
9.2. User Backend.....	36
9.3. User Auth Backend.....	36
9.3.1. Database (default)	36
9.3.2. LDAP	36
10. Customer (min. OTRS 1.1).....	38
10.1. How it works	38
10.1.1. Agent-Interface	38
10.1.2. Admin-Interface.....	38
10.1.3. Customer-Interface.....	38
10.2. Customer User Backend	38
10.2.1. Database (default)	38
10.2.2. LDAP	39
10.3. Customer Auth Backend	40
10.3.1. Database (default)	40
10.3.2. LDAP	40
10.4. Customize Customer Self Registration	40
10.4.1. Template.....	40
10.4.2. Customer Map.....	41
10.4.3. Modify Customer Table	41
11. Permissions	42
11.1. Groups - Queue.....	42
11.2. Agent - Admin Interface	42
12. Config File.....	43
12.1. TicketHook.....	43
12.2. FQDN.....	43
12.3. Log	43
12.4. Check Email Options	43
12.5. Max. Postmaster email.....	44
12.6. Ticket number format.....	44
12.7. Database settings.....	45
12.8. ASP (Application Service Provider) options	45
12.9. Custom Queue.....	45
12.10. Agent notification.....	46
12.11. Session management.....	46
12.12. URL login and logout settings	47
12.13. agent area default settings	47
12.14. SpellChecker	49

13. Cronjobs.....	50
13.1. How it works - setup	50
13.2. Default cronjobs.....	50
13.2.1. aaa_base	50
13.2.2. pending_jobs	50
13.2.3. postmaster	50
13.2.4. postmaster_pop3	50
13.2.5. unlock.....	50
13.2.6. stats	51
13.2.7. session.....	51
13.2.8. rebuild_ticket_index.....	51
13.2.9. generic_agent	51
14. Generic-Agent	52
14.1. Config File	52
14.2. Examples.....	52
14.2.1. close spam tickets	52
14.2.2. delete tickets.....	53
14.2.3. move tickets from tricky to experts.....	53
14.2.4. move escalation ticket to experts and execute CMD	54
15. Language translations	55
15.1. New translation files	55
16. Customize the frontend	57
16.1. The dtl syntax.....	57
16.2. Examples of dtl files.....	58
16.2.1. Motd.dtl.....	58
16.2.2. Login.dtl.....	58
16.2.3. Header.dtl	59
16.3. Create a new theme	60
17. Customize Ticket State (min. OTRS 1.1).....	61
18. Customize Ticket Priority (min. OTRS 1.1)	62
19. Multi Hosting.....	63
19.1. Multi Group	63
19.2. Multi Setup	63
20. Performance Tuning	65
20.1. OTRS	65
20.1.1. TicketIndexModule	65
20.1.2. TicketStorageModule.....	65
20.2. Database.....	65
20.3. Webserver.....	65
21. Troubleshooting.....	67
21.1. General problems with OTRS on SuSE Linux	67
21.1.1. SuSE Linux 8.0	67
21.1.2. SuSE Linux and Postfix	67
21.2. General problems with OTRS on other distributions (e.g. Redhat).....	67
21.3. Problems with Apache	67
21.3.1. Internal Server Error	67
21.3.2. Error: Can't connect to database!	68
21.3.3. FreeBSD, PostgreSQL and Apache (install_driver(Pg) failed).....	68
21.4. Problems with MySQL	68
21.4.1. Check the database connect	68
21.4.2. Access denied for user: 'otrs@localhost'	68
21.4.3. Lost root password of MySQL	69
21.4.4. http://localhost/otrs/installer.pl	69
21.5. Problems - misc.....	69
21.5.1. Problems with receiving emails	69
21.5.2. Lost root password of OTRS	69

22. FAQ.....	70
Glossary	73
A. Online resources	76
A.1. Homepage	76
A.2. Mailinglists	76
A.3. Bugtracking	76
B. The OTRS core team.....	78
B.1. Martin Edenhofer	78
B.2. Stefan Wintermeyer.....	78
B.3. Sebastian Wormser	78
C. Credits	79
D. RFC 1297	80
E. GNU Free Documentation License	81
0. PREAMBLE	81
1. APPLICABILITY AND DEFINITIONS	81
2. VERBATIM COPYING	81
3. COPYING IN QUANTITY	82
4. MODIFICATIONS	82
5. COMBINING DOCUMENTS.....	83
6. COLLECTIONS OF DOCUMENTS	83
7. AGGREGATION WITH INDEPENDENT WORKS	83
8. TRANSLATION	83
9. TERMINATION	84
10. FUTURE REVISIONS OF THIS LICENSE.....	84
How to use this License for your documents	84
Bibliography	85

List of Examples

7-1. .procmailrc	32
7-2. .fetchmailrc	34
12-1. Kernel/Config.pm - TicketHook	43
12-2. Kernel/Config.pm - FQDN	43
12-3. Kernel/Config.pm - LogModule	43
12-4. Kernel/Config.pm - CheckEmailAddresses	??
12-5. Kernel/Config.pm - CheckMXRecord	44
12-6. Kernel/Config.pm - PostmasterMaxEmails	44
12-7. Kernel/Config.pm - TicketNumberGenerator	44
12-8. Kernel/Config.pm - Database settings	45
12-9. Kernel/Config.pm - MoveInToAllQueues	45
12-10. Kernel/Config.pm - ChangeOwnerToEveryone	45
12-11. Kernel/Config.pm - Custom Queue	45
12-12. Kernel/Config.pm - Agent notification	46
12-13. Kernel/Config.pm - Session management	46
12-14. Kernel/Config.pm - URL login and logout settings	47
12-15. Kernel/Config.pm - agent area default settings	48
12-16. Kernel/Config.pm - SpellChecker	49
12-17. Kernel/Config.pm - SpellChecker - Agent Preferences	49
12-18. Kernel/Config.pm - SpellChecker - Must be spell checked	49
15-1. Kernel/Language/de.pm - Old file	55
15-2. Kernel/Language/fr.pm - New file	56

Foreword

About this Book

This is an alpha edition of this book. This version may contain glaring inconsistencies, missing sections, and other misfeatures indicative of a work in progress. But please do not hesitate to add every found mistake in our bugtracking tool at <http://bugs.otrs.org/>.

You will find the current HTML online version of this book at <http://otrs.org/docu/> or in PDF-format at <http://otrs.org/docu/manual.pdf>.

Anyhow we are keen on your feedback. Please do not hesitate to send us an e-mail to <feedback@otrs.org>

Your OTRS core team

Chapter 1. Install the biest.

The described way of installing the otrs is tested on a newly installed system. In case you have trouble to install it on your working system please try it with a new Linux installation. Most of the installation problems are caused by messy apache configurations and forgotten mysql database passwords. So please use a fresh installation to encircle an installation problem in case it occurs.

We try to keep the installation as easy as possible. But the OTRS is a very powerfull and complex application which can not just be untared in some directory and that's it.

1.1. Installing the rpm on a SuSE Linux (the quick and easy way)

This section is a guide for installing OTRS on a SuSE Linux. We tested the SuSE Linux versions 7.3, 8.0 and 8.1. Before starting the installation have a look at <http://otrs.org/> (<http://www.otrs.org/>) and check if a newer and better version of the rpm file is available. If so please download it and use the newer documentation and the newer rpm.

Install the otrs.rpm with YaST (YaST2) or on the command line with rpm (what ever you prefer). Please be aware of the fact that OTRS needs some Perl-Modules which are not installed by default in a typical SuSE installation. So it might be a good idea to use YaST to install the rpm because it will handle and solve all the dependencies.

In case you prefer the command line rpm way (you have to have installed the needed modules first otherwise rpm will ask you to do so):

```
ernie:~ # rpm -ivh /tmp/otrs.rpm
otrs                                     #####
Check OTRS user (/etc/passwd)... otrs exists.

Next steps:

[SuSEconfig]
  Execute 'SuSEconfig' to configure the webserver.

[start Apache and MySQL]
  Execute 'rcapache restart' and 'rcmysql start' in case they don't run.

[install the OTRS database]
  Use a webbrowser and open this link:
  http://localhost/otrs/installer.pl

[OTRS services]
  Start OTRS 'rcotrs start-force' (rcotrs {start|stop|status|restart|start-force|stop-force}).

Have fun!

  Your OTRS Team
  http://otrs.org/

ernie:~ #
```

Know it's time to start SuSEconfig:

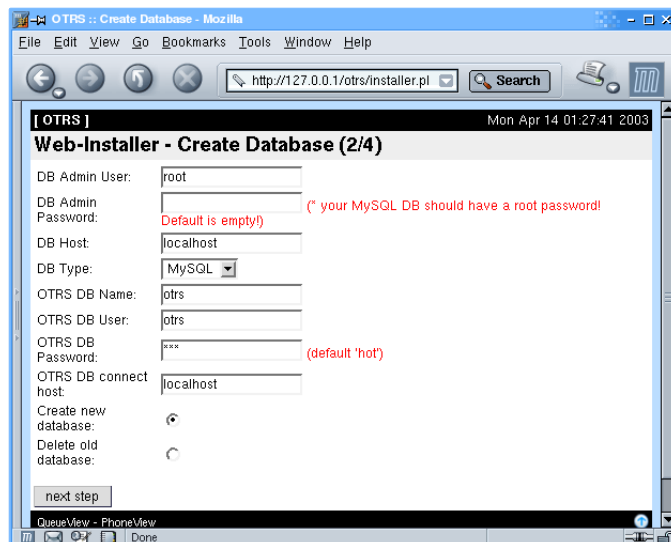
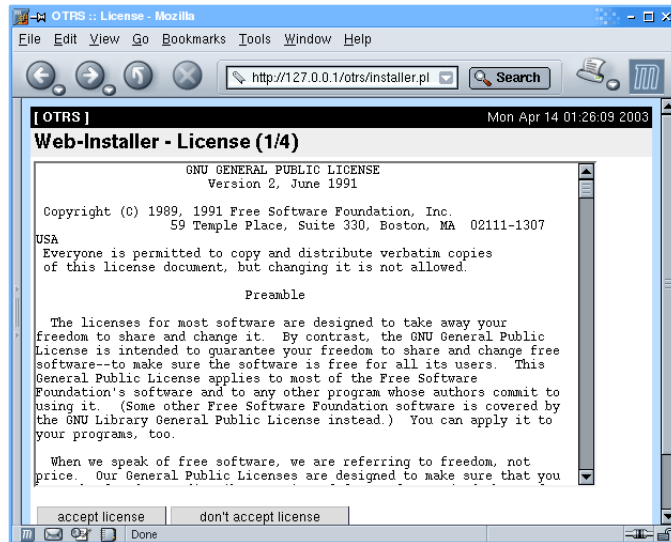
```
ernie:~ # SuSEconfig
Starting SuSEconfig, the SuSE Configuration Tool...
Running in full featured mode.
Reading /etc/sysconfig and updating the system...
Executing /sbin/conf.d/SuSEconfig.aaa_at_first...
Executing /sbin/conf.d/SuSEconfig.apache...
Including /opt/otrs/scripts/apache-httpd.include.conf
Executing /sbin/conf.d/SuSEconfig.bootsplash...
Executing /sbin/conf.d/SuSEconfig.doublecheck...
Executing /sbin/conf.d/SuSEconfig.guile...
Executing /sbin/conf.d/SuSEconfig.hostname...
Executing /sbin/conf.d/SuSEconfig.ispell...
Executing /sbin/conf.d/SuSEconfig.perl...
Executing /sbin/conf.d/SuSEconfig.permissions...
Executing /sbin/conf.d/SuSEconfig.postfix...
Setting up postfix local as MDA...
Setting SPAM protection to "off"...
Executing /sbin/conf.d/SuSEconfig.profiles...
Finished.
```

Once you installed the `otrs.rpm` on your system you have to restart the apache by `rcapache restart` to force apache to reload the config file.

```
ernie:~ # rcapache restart
Shutting down httpd
Starting httpd [ PERL ]
ernie:~ #
```

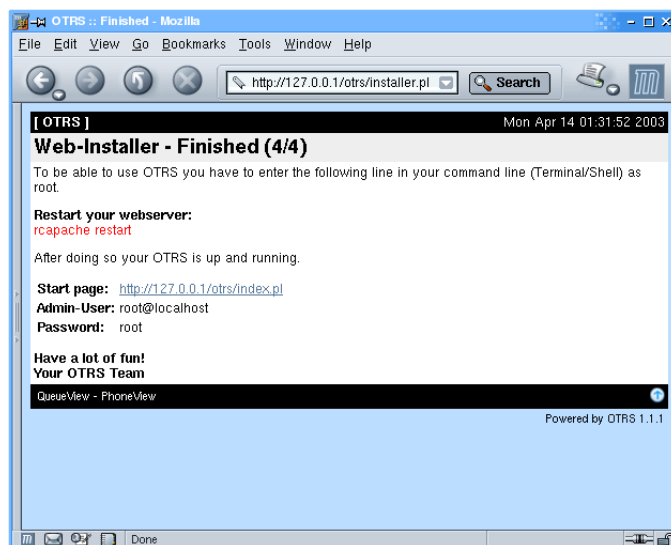
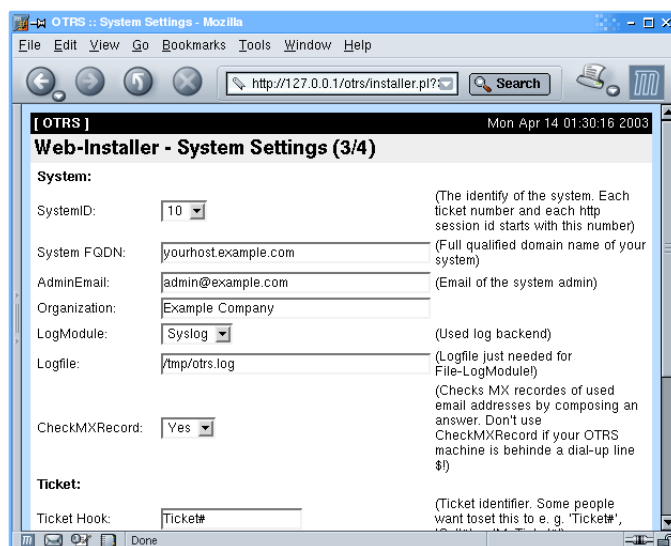
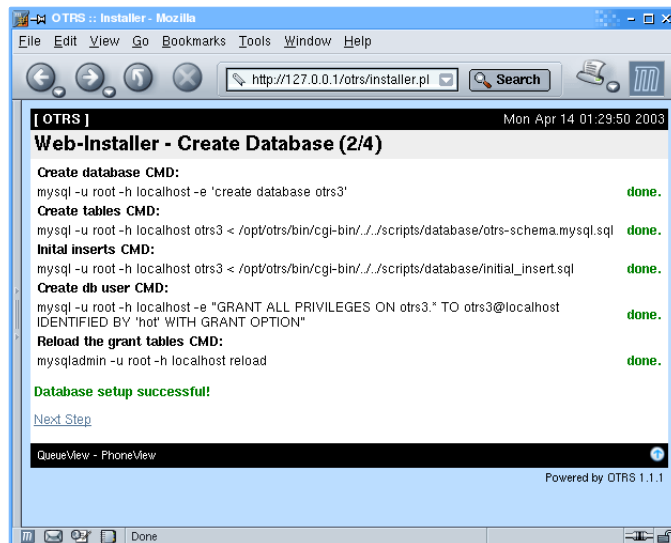
done
done

Now you have to setup the databases. Please open <http://localhost/otrs/installer.pl> to do so.



Warning

It is not a good idea to use the default passwords. You install a very important piece of software and you do not want anybody to be able to hack your database just because you didn't change the default password!



OK, now it is time to fire up the otrs. Do so on the command line:

```
ernie:~ # rcotrs restart-force
Shutting down OTRS
  Disable /opt/otrs/bin/PostMaster.pl ... done.
no crontab for otrs
  Shutting down cronjobs ... failed!
Shutting down OTRS (completely)
  Shutting down Apache ... done.
  Shutting down MySQL ... done.
done

Starting OTRS (completely)
  Starting Apache ... done.
  Starting MySQL ... done.
Starting OTRS
  Checking Apache ... done.
  Checking MySQL ... done.
  Checking database connect... (It looks Ok!).
  Enable /opt/otrs/bin/PostMaster.pl ... done.
  Checking otrs spool dir... done.
  Creating cronjobs (source /opt/otrs/var/cron/*) ... done.

-->> http://ernie.example.com/otrs/index.pl <--
done
done

ernie:~ #
```

Finish! Wasn't that a piece of cake? ;-)

You can use the OTRS by opening the <http://ernie.example.com/otrs/index.pl> link.

1.2. Using the tar.gz file to install OTRS on any Linux/Unix platform

This section is a guide for installing OTRS on any Linux. Please use this way only in case you feel comfortable with it otherwise use the RPM. Before starting the installation have a look at <http://otrs.org/> (<http://www.otrs.org/>) and check if a newer and better version of the tar.gz file is available. If so please download it and use the newer documentation and the newer rpm.

1.2.1. Install

```
Software requirements?
=====

On all Perl-Platforms! You need:
* min. Perl5
* Database (e. g. MySQL, PostgreSQL)
* Webserver (mod_perl isn't required but "very nice" to have)
* LDAPv2 compliant server (OpenLDAP for works fine, however LDAP isn't required)
* some CPAN-Module (DBI, DBD::mysql, Digest::MD5, MIME::Base64 (if Perl < 5.8), Net::DNS)
  for LDAP (Net::LDAP) and for stats (GD, GD::Text, GD::Graph, GD::Graph::lines,
  GD::Text::Align)

Installation:
=====

This few steps describe an OTRS installation incl. webserver and database
settings. The OTRS user is "otrs" and the home (root) directory is /opt/otrs
(of cause you can choose an other directory or/and OTRS user).

1. Install CPAN Modules (if needed):
-----

Note: use "bin/otrs.checkModules" to get an overview of all installed and
required cpan modules.

a) Install the RPMs if your distributions provides RPMs for
```

the required CPAN modules.

```
b) Install the required modules via CPAN shell (http://www.cpan.org/)
perl -MCPAN -e shell;
...
install Digest::MD5
install Net::DNS
install MIME::Base64 (if Perl < 5.8)
...

if you plan to use an LDAP directory service you should install Net::LDAP
...
install Net::LDAP
...
and maybe the GD stuff (stats support, not required!)
...
install GD
install GD::Text
install GD::Graph
install GD::Graph::lines
install GD::Text::Align
...
```

Check if all needed modules are installed:

```
-----
$shell:~> perl -cw /opt/otrs/bin/cgi-bin/index.pl
/opt/otrs/bin/cgi-bin/index.pl syntax OK
$shell:~> perl -cw /opt/otrs/bin/PostMaster.pl
/opt/otrs/bin/PostMaster.pl syntax OK
$shell:~>
```

If you get "syntax OK" it seems to be Ok. Go ahead.

2. Create user:

```
-----
Add user:
-----
$shell: useradd -d /opt/otrs/ -c 'OTRS user' otrs

Add user to webserver group (if the webserver is not running with OTRS user):
-----
$shell: usermod -G nogroup otrs

(SuSE=nogroup, Red Hat=apache)
```

3. Install tar.gz:

```
-----
$shell: cd /opt/
$shell: tar -xvzf otrs-xxxx-xx-xx.tar-gz
```

4. Demo config files:

```
-----
There are several OTRS demo config files in $OTRS_HOME/Kernel/*.dist
and $OTRS_HOME/Kernel/Config/*.dist. Make copies of all demo config files:
```

```
cp Kernel/Config.pm.dist Kernel/Config.pm

cd Kernel/Config/
for foo in *.dist; do cp $foo `basename $foo .dist`; done
```

Or if you are installing OTRS on a Windows system:

```
copy Kernel/Config.pm.dist Kernel/Config.pm

cd Kernel/Config/
copy *.dist *
```

5. Webserver:

```
-----
[follow README.webserver]
```

6. File Permissions:

```
-----  
Set the file permissions with  
"$HOME/bin/SetPermissions.sh <OTRS_HOME> <OTRS_USER> <WEBSERVER_USER> [OTRS_GROUP] [WEB_GROUP]"  
e. g.  
  Webserver with OTRS user: "SetPermissions.sh /opt/otrs otrs otrs"  
  Webserver with wwwrun user (e. g. SuSE): "SetPermissions.sh /opt/otrs otrs wwwrun"  
  Webserver with apache user (e. g. Redhat): "SetPermissions.sh /opt/otrs otrs apache apache apache"
```

7. Database setup:

```
-----  
If you use MySQL, you can use the Web-Installer (http://yourhost/otrs/installer.pl).  
Else follow README.database --> "DB - Setup Example".
```

8. Config file (\$HOME/Kernel/Config.pm):

```
-----  
If you used the Web-Installer, you can skip this point. If not,  
set some Kernel::Config ($HOME/Kernel/Config.pm) Options (FQDN, SystemID, TicketHook,  
Home, ...)
```

9. First Login:

```
-----  
http://yourhost/otrs/index.pl  
User: root@localhost  
PW: root  
  
--> goto AdminArea and set some config settings (UserAdd, Queues, ...).  
  
Finished.
```

10. First Email:

```
-----  
pipe an email directly into $OTRS_HOME/bin/Postmaster.pl  
(e. g. 'cat /opt/otrs/doc/test-email-1.box | /opt/otrs/bin/PostMaster.pl').
```

11. Cronjobs for the OTRS user:

```
-----  
There are several OTRS default cronjobs in $OTRS_HOME/var/cron/*.dist.  
Make copies of all of the default cronjobs:
```

```
cd var/cron  
for foo in *.dist; do cp $foo `basename $foo .dist`; done
```

Or if you are installing OTRS an a Windows system:

```
cd var/cron  
copy *.dist *.
```

Use \$OTRS_HOME/bin/Cron.sh {start|stop|restart} to start or stop this cronjobs
from \$OTRS_HOME/var/cron/* (.dist will be ignored).

>> Note: Install this cronjobs as OTRS user. <<

Remark: For installation questions ask otrs@otrs.org (<http://lists.otrs.org/>).

Have a lot of fun.

Martin Edenhofer

(January 2003 Frankfurt/Germany)

EOF

1.2.2. Database

Where can I find the database description files?
=====

XML:
====
\$HOME_OTRS/scripts/database/otrs-schema.xml

The XML description files for torque which generate the SQL for your target database e. g. MySQL, PostgreSQL, DB2, Oracle, ...)

More Infos: <http://jakarta.apache.org/turbine/turbine-2/howto/torque-howto.html>

MySQL:

\$HOME_OTRS/scripts/database/otrs-schema.mysql.sql

PostgreSQL:

\$HOME_OTRS/scripts/database/otrs-schema.postgresql.sql

Initial insert file:
=====

\$HOME_OTRS/scripts/database/initial_insert.sql contains all needed standard values. At first use the otrs-schema.*.sql and the insert this file.

DB - Setup Example (MySQL):
=====

Create OTRS database:

```
shell> mysql -u root -p -e 'create database otrs'
```

Create the OTRS tables:

```
shell> mysql -u root -p otrs < scripts/database/otrs-schema.mysql.sql
```

Insert initial data:

```
shell> mysql -u root -p otrs < scripts/database/initial_insert.sql
```

Create an database user:

```
shell> mysql -u root -p -e 'GRANT ALL PRIVILEGES ON otrs.* TO otrs@localhost IDENTIFIED BY "some-pass" WITH GR
```

Reload the grant tables of your mysql-daemon:

```
shell> mysqladmin -u root -p reload
```

```
*****
*
* Change the DB-Settings (host, database, user and password) *
*
* $OTRS_HOME/Kernel/Config.pm
* [...]
* # Database
* # (The database name.)
* $Self->{Database} = 'otrs';
*
* # DatabaseUser
* # (The database user.)
* $Self->{DatabaseUser} = 'otrs';
*
* # DatabasePw
* # (The password of database user.)
* $Self->{DatabasePw} = 'some-pass';
* [...]
*
*****
```

EOF

1.2.3. Webserver

```
Which webserver is needed?
=====
I prefer the apache webserver (http://httpd.apache.org).

Configuration:
=====
"After" this steps, you will get the login page at
http://your-host/otrs/index.pl or http://your-host/otrs/installer.pl.

SuSE Linux:
=====
a) Install the RPM-Package (http://otrs.org/ - "rpm -i otrs-xxx.rpm").

b) The manual way:
   Use the "$OTRS_HOME/scripts/suse-httpd.include.conf" include config file.

   Add it to /etc/sysconfig/apache with HTTPD_CONF_INCLUDE_FILES
   [...]
   HTTPD_CONF_INCLUDE_FILES=/opt/otrs/scripts/suse-httpd.include.conf
   [...]

   Start SuSEconfig and restart the webserver (rcapache restart).

Or edit the httpd.conf directly:
=====
a)
*) Change the webserver user (normaly wwwrun) to the OTRS user (otrs).

   [...]
   User wwwrun
   [...]
   User otrs
   [...]

*) If you can't change the user and group of your webserver (system-wide),
   because you have other applications running on this server, you can
   also work with group permissions (more tricky).

   Use "$HOME/bin/SetPermissions.sh <OTRS_HOME> <OTRS_USER> <WEBSEVER_USER> [OTRS_GROUP] [WEB_GROUP]"
   e. g.
   Webserver with otrs user: "SetPermissions.sh /opt/otrs otrs otrs"
   Webserver with wwwrun user (e. g. SuSE): "SetPermissions.sh /opt/otrs otrs wwwrun"
   Webserver with apache user (e. g. Redhat): "SetPermissions.sh /opt/otrs otrs apache"

b)
*)
   Without mod_perl (just CGI):
   =====
   Add this to the cgi-bin stuff section in httpd.conf
   [...]
   Alias /otrs-web/ "/opt/otrs/var/httpd/htdocs/"
   ScriptAlias /otrs/ "/opt/otrs/bin/cgi-bin/"
   [...]

*)
   With mod_perl (speed!):
   =====
   Add this to the mod_perl stuff section in httpd.conf
   [...]
   Alias /otrs-web/ "/opt/otrs/var/httpd/htdocs/"
   Alias /otrs/ "/opt/otrs/bin/cgi-bin/"
```

```
<Location /otrs>
    SetHandler perl-script
    PerlHandler Apache::Registry
    Options ExecCGI
    PerlSendHeader On
</Location>
```

You may want to use a `mod_perl` startup script. Compiled modules on startup (speed!!) Use the `mod_perl` startup script which comes with `otrs` (`scripts/apache-perl-startup.pl`).

```
- Change the default startup script location of your httpd to
$OTRS_HOME/scripts/apache-perl-startup.pl in httpd.conf
[...]
# load all otrs modules
PerlRequire /opt/otrs/scripts/apache-perl-startup.pl
[...]
```

Edit the `scripts/apache-perl-startup.pl` script:

```
- Establish datababase connections on process startup (httpd).

[...]
use Apache ();
use Apache::DBI ();
Apache::DBI->connect_on_init('DBI:mysql:otrs', 'otrs', 'some-pass');
# Apache::DBI->connect_on_init($data_source, $username, $auth, \%attr)
[...]

- Change the otrs lib dir (if needed)!

[...]
# --
# set otrs lib path!
# --
use lib "/path/to/otrs/";
use lib "/path/to/otrs/Kernel/cpan-lib";
[...]
```

Nice! You will love `mod_perl`! , -)

PS: If you use `mod_perl2` use the `scripts/apache2-*` scripts!

c)

```
Restart the webserver
=====
```

d)

```
Web-Installer
=====
http://yourhost/otrs/installer.pl
```

```
First login
=====
http://yourhost/otrs/index.pl
User: root@localhost
PW: root
```

EOF

1.3. Install OTRS on Win32

This How-To installs the Open Ticket Request System OTRS(<http://otrs.org>) on systems running *Win32*. I tested it on Windows 2000, it should run on Windows 98, 98SE, ME, NT4 and XP, too.

Command Boxes: Throughout this document, we will need to enter commands directly into the system quite often. We do this in a so called *command box*. Other names are *command line*, (*DOS*) *prompt* or *shell*. Here is how to invoke a command box on Windows:

Click *Start*, then *Execute* and enter **cmd** in the dialogue box popping up. Hit **Return** on your keyboard or click *OK*. A (usually black) box appears, showing something similar to this:

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
```

```
C:\>
```

You may now enter commands such as

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
```

```
C:\>cd /d d:\otrs\bin\cgi-bin
```

```
D:\otrs\bin\cgi-bin>dir
Datenträger in Laufwerk D: ist Aurelia
Datenträgernummer: C8F1-F408
```

```
Verzeichnis von D:\otrs\bin\cgi-bin
```

```
20.01.03  00:20          <DIR>          .
20.01.03  00:20          <DIR>          ..
08.06.02  23:37             10 .cvsignore
18.01.03  02:14             20 .htaccess
09.01.03  16:06          20.734 customer.pl
09.01.03  16:06          19.014 index.pl
03.01.03  17:17           4.203 installer.pl
03.01.03  17:17           5.450 pic.pl
           6 Datei(en)          49.431 Bytes
           2 Verzeichnis(se),   9.825.726.464 Bytes frei
D:\otrs\bin\cgi-bin>
```

This will bring you to your OTRS executable directory and show it's contents. The */d* switch tells Windows to also change the drive letter, if necessary, i.e. to directly switch to the given directory.

Please note that this is a *sample output only*, and it's from my box, a german Windows 2000 installation, as you see. It will probably look different on your box.

You can customize the appearance of command boxes by clicking in the upper left corner of the window and choose **Properties**.

It will help if you get a little

bit comfortable with command boxes and working from the command line in general. Search with Google for help with command lines(<http://www.google.com/search?num=100&newwindow=1&q=%22command+line%22+%22learning+resources%22+windows>).

1.3.1. Installations

You will need three main components on your system, a webserver (Sambar Server(<http://www.sambar.com/>) or/and Apache(<http://apache.org/>)), a database (MySQL(<http://www.mysql.com/>)) and the programming language Perl(<http://www.perl.org/>).

The whole installation will assumingly take place on drive D:, but you may install wherever you like, for sure. All you have to do is replace every occurence of *D:* or *d:* with your choice, f.e. *c:\Programs*. It is **no** good idea to have spaces in the installation path, so *c:\program Files\otrs* won't be a lucky choice.

1.3.1.1. Install PERL

Download *Perl, Version 5.6.1 build 633* from <http://www.activestate.com/Products/Download/Download.plex?id=ActivePerl>. I recommend choosing the MSI version.

Install Perl using the default values to *D:\Perl*.

1.3.1.1.1. Install Additional PERL Packages

Download the OTRS-Win32-Perl-Packages in one file (<http://robert.kehl.bei.t-online.de/OTRS-Win32-Perl-Packages.zip>) (coming from *robert.kehl.bei.t-online.de*, not browsable) and unpack the file into *D:\Perl*, preserving directory structure.

1.3.1.2. Choosing And Installing The Webserver

OTRS requires a minimum of one webserver to run ;) You may choose the one of your choice, it will probably run on every perl-enabled webserver. I tested it on Sambar (<http://www.sambar.org>) 5.2 & above (<http://sambar.robertkehl.de>) and on Apache (<http://httpd.apache.org/>), both 1.3 (<http://httpd.apache.org/docs/>) and 2.0 (<http://httpd.apache.org/docs-2.0/>). *mod_perl* is only available for Apache. Currently, only *mod_perl-1.0* on Apache 1.3.27 is working ok for OTRS, *mod_perl-2.0* regrettably not.

1.3.1.2.1. Installing Apache

Download the Apache (<http://www.apache.org>) of your choice (1.3.27 (http://www.apache.org/dist/httpd/binaries/win32/apache_1.3.27-win32-x86-no_src.msi) and/or 2.0.43 (http://www.apache.org/dist/httpd/binaries/win32/apache_2.0.43-win32-x86-no_ssl.msi)) and install it/them. I recommend you choose **D:\Apache** as root for *both* versions. The resulting directory structure will be:

```
D:\Apache\          --- home of both
D:\Apache\Apache\   --- home of Apache 1.3
D:\Apache\Apache2\  --- home of Apache 2.0
```

Tip: You may install both 1.3 and 2.0 on the very same machine, they can both be running and may both be used at the same time, if you like to. All you have to do is to ensure they are not configured to run on the same port. The second installation willing to start on port 80 won't succeed in doing so. So configure Apache 1.3 to run on port 80, Apache 2.0 to run on port 81. You could even configure the Sambar webserver (see different section in this manual) to run on port 82, if you'd like to, or choose your own port configuration. Surely one webserver running one version would be enough - the rest is for geeks ;)

1.3.1.2.2. Install Sambar Server

Download Sambar Server from <http://sambar.robertkehl.de>, the home page is <http://www.sambar.com>. I recommend using version 5.2 Production, surely any later version will do, too. Do not use any version prior to 5.2 Production!

Read the security notes on <http://www.sambar.com/syshelp/security.htm> and install Sambar, where you like to, *D:\Sambar* is a good choice.

If you're running Windows NT4, 2000 or XP, you can and should install Sambar as a Service, if the install routine hasn't done so yet. Therefore you open a command box. You enter the following:

```
C:\>cd /d d:\sambar\bin
D:\sambar\bin>ntserver.exe -i -s Sambar
```

Now start Sambar using the built-in service administration panel or by entering:

```
D:\sambar\bin>net start Sambar
```

On Win9x/ME, you start Sambar clicking *Start -> Programs -> Sambar Server -> Start Sambar Server*. There are no services on Win9x/ME.

Ensure that Sambar is running: <http://localhost>. The documentation can be found here: <http://localhost/syshelp/index.htm>. The system administration forms are here: <http://localhost/session/adminlogin?RCpage=/sysadmin/index.stm>.

Important: Ensure to *now* set the admin password to prevent others using your Sambar in a malicious way. You do this on the sysadmin forms under *User Management*. Good Passwords are longer than eight keystrokes and look like this:
sie.&Ph9w_iG

1.3.1.3. Install MySQL

Download MySQL from <http://www.mysql.com/downloads/mysql-3.23.html> and install it in *D:\mysql*, under Win 2000/XP as a service, too. Start it.

For a more comfortable configuration interface I recommend installing phpMyAdmin from <http://www.phpmyadmin.net>, too, location: *D:\mysql\phpMyAdmin*. This will require PHP(<http://www.php.net>) to be installed, which always is a very good idea.

We change the password for root later.

1.3.1.4. Install OTRS

Last but not least - the beast!

Download OTRS as a tarball (.tar.gz): <http://otrs.org/pages/index.pl?Site=Download>.

We have to install this tarball manually, because there is a file "INSTALL" in it, who has the same name like the directory "install", which is also contained. Un*x differentiates between the two - Windows doesn't. In addition, the three files .fetchmailrc, .mailfilter and .procmailrc are doubled, your unzip program will complain about this. Future releases will allow automatic unzipping.

Open the tarball using WinRAR (<http://www.rarlab.com>), Winzip (<http://www.winzip.com>) or a similar software. Unzip the files .fetchmailrc, .mailfilter, .procmailrc and INSTALL to *d:\otrs*. Rename INSTALL to INSTALL.txt. Now unzip the rest of the archive to *d:\otrs*.

1.3.2. Configuration

1.3.2.1. Configuring Perl

We now patch our Perl because it doesn't suit our needs in the default installation, there are some so called "packages" missing.

Open a command box and enter this:

```
C:\>cd /d D:\Perl\packages

D:\Perl\packages>install.bat
[...returned output shortened...]
```

The install routine places the file *mod_perl.so* in **\Apache\Apache\modules**. By the time of this writing, you will have to manually adjust the path in *install.bat*, if you installed Apache in another place or copy *mod_perl.so* manually to your Apache's modules directory.

Your Perl should now look like this or even better:

```
D:\Perl\packages>ppm query
[...returned output shortened...]
Archive-Tar      [0.072    ]
Authen-SASL      [2.03     ]
Compress-Zlib    [1.16     ]
Convert-ASN1     [0.16     ]
DBD-MySQL        [1.2200   ]
DBI              [1.27     ]
Digest           [1         ]
Digest-HMAC      [1.01     ]
Digest-MD2       [2         ]
Digest-MD4       [1.1      ]
Digest-MD5       [2.20     ]
Digest-SHA1      [2.01     ]
```

```

File-CounterFile [0.12      ]
Font-AFM         [1.18      ]
GD               [1.27.2    ]
GDGraph          [1.32      ]
GDTextUtil       [0.80      ]
HTML-Parser      [3.26      ]
HTML-Tagset      [3.03      ]
HTML-Tree        [3.11      ]
IO-Socket-SSL    [0.92      ]
IO-stringy       [2.108     ]
MD5              [2.02      ]
MIME-Base64      [2.12      ]
MIME-tools       [5.411a     ]
MailTools        [1.58      ]
Net-DNS          [0.33      ]
Net_SSLeay.pm    [1.22      ]
PPM              [2.1.6     ]
SOAP-Lite        [0.55      ]
Storable         [1.0.12    ]
Test-Simple      [0.47      ]
Tk               [800.023    ]
URI              [1.19      ]
XML-Parser       [2.27      ]
XML-Simple       [1.06      ]
libnet           [1.12      ]
libwin32         [0.19.1    ]
libwww-perl      [5.64      ]
mod_perl         [1.27_01-dev]
perl-ldap        [0.26      ]

```

You may now delete the files in D:\Perl\packages, if you want to, but you don't have to.

That's it for Perl.

1.3.2.2. Configuring The Webserver

Again we differentiate between Apache and Sambar. Remember you can have both running or switch around if you like.

1.3.2.2.1. Configuring Apache

In the following, I will refer to 'Apache' only, the process is the same for both versions 1.3 and 2.0.

Open Apache's configuration file **httpd.conf**, it is located in **d:\Apache\Apache\conf**. Append these lines to the end of the file, adjust them to your needs:

```

# uncomment the following two for Apache2!
LoadModule perl_module modules/mod_perl.so
AddModule mod_perl.c

### added for OTRS (http://otrs.org/)

<IfModule mod_alias.c>
Alias /otrs/ "d:/otrs/bin/cgi-bin/"
PerlModule Apache::Registry
</IfModule>

<Location /otrs>
Options ExecCGI
Order deny,allow
Deny from all
allow from 127.0.0.1

SetHandler cgi-script
ScriptInterpreterSource registry

<IfModule mod_perl.c>
SetHandler perl-script
PerlHandler Apache::Registry
PerlSendHeader On

```

```

</IfModule>

</Location>

# load all otrs modules
PerlRequire d:/otrs/scripts/apache-perl-startup.pl

# MaxRequestsPerChild (so no apache child will be to big!)
#MaxRequestsPerChild 400
# depends on your RAM

```

Take a little time to configure the rest of httpd.conf, too. Open d:\otrs\scripts\apache-perl-startup.pl and change these lines:

```

1:
#! D:/Perl/bin/perl
10,11:
use lib "d:/otrs/";
use lib "d:/otrs/Kernel/cpan-lib";
38,39:
#use Kernel::System::AuthSession::IPC;
use Kernel::System::AuthSession::DB;
46,47:
#use Kernel::System::Log::SysLog;
use Kernel::System::Log::File;

```

About mod_perl_2.0 on Apache2: At this moment, OTRS under mod_perl_2.0 isn't running smoothly on Win32, so perl is used as a CGI process on Apache2 only. We are fully aware that mod_perl_2.0 should be used to efficiently speed things up, and we're in the process of getting it to work. Please be patient or try mod_perl yourself. Let us know(mailto:otrs-win32@robertkehl.de?subject?mod_perl) if you succeed :)

That's it for Apache. (Re-)Start your installation(s) using the Apache monitor, probably located in your task bar. Sometimes it's a good idea to restart the whole box.

1.3.2.2.2. Configuring Sambar

We now configure our webserver *Sambar*. Open the file *D:\Sambar\config\mappings.ini* and create these entries:

```

[aliases]
/mysql = d:/mysql/Docs
/phpMyAdmin = d:/mysql/phpMyAdmin

[cgi-aliases]
/otrs/ = /otrs/bin/cgi-bin/
/obin/ = /otrs/bin/

```

In *D:\Sambar\config\config.ini* you set values shown beneath::

```

[common]
Trace Level = INFO
Trace Performance = true
Network Trace Level = None
Dynamic IP Test = true
License =
Licensee =
DNS Primary = IPAdresse.DNS1.beideinem.Provider
DNS Secondary = IPAdresse.DNS2.beideinem.Provider

[server]
System Administrator = admin
System Administrator IP = 127.0.0.1
Act As HTTP Server = true
Act As HTTPS Server = false

```

```

Act As DNS Server = false
Act As DHCP Server = false
Act As Mail Server = false
Act As TFTP Server = false
Act As FTP Server = false
Act As FTPS Server = false
Act As FTP Proxy = false
Act As NNTP Proxy = false
Act As SMTP Proxy = false
Act As POP3 Proxy = false
Act As IMAP4 Proxy = false
Act As Bridge Proxy = false
Act As SOCKS Proxy = false
Act As Telnet Server = false
Trace FTP = true
Trace TFTP = true
Trace Bridge = true
Trace DHCP = true
Trace DNS = true
SMTP Server = smtp.deinprovider.de
DOT-File Security = true

[http]
Act As HTTP Proxy = false
Trace Proxy Usage = true
Trace Requests = true
Log Format = performance
Don't Log IPs = 127.0.0.1
Default Page = index.pl index.php index.stm index.shtml index.shtm index.sht index.htm index.html
Perl Executable = D:/Perl/bin/perl.exe
CGI Extensions = *.pl
Enforce .htaccess = true

[events]
Monitor Invalid Logins = true
Monitor Invalid Requests = true
Share User Logins = true
Trace User Logins = true

```

Restart Sambar for the changes to take affect.

Note:: At the very moment we do not use the *mail server* of Sambar, nor the *DNS server*. To use them right now (outside of OTRS), you need a *Pro version* of Sambar, see <http://sambar.com/syshelp/pro.htm>. The http functionality is not limited in the freeware version, which is automagically active. Entering this in config.ini will enable the demo version of the Pro version aktiviert, which will run between 9 a.m. and 5 p.m.

```

[common]
License = demo
Licensee = demo

```

1.3.2.2.2.1. Securing /otrs/bin

We **have** to secure the alias /obin/ which points to d:\otrs\bin\, holding all of the code useful for hacking the system. I personally do not want to expose these "tools" to everyone.

Create the file *d:\otrs\bin\htaccess.txt* using Windows Explorer containing just this one line:

```
require group root
```

Open a command box and enter this:

```
C:\>cd /d d:\otrs\bin
```

```
D:\otrs\bin>ren htaccess.txt .htaccess
```

Now only users belonging to the group "root" and the system itself (CRON jobs f.e.) will be allowed to access the scripts in "d:\otrs\bin".

1.3.2.3. Configuring MySQL

Secure your MySQL by providing good passwords for the users @%, root@% und root@localhost. This can easily be done via *phpMyAdmin*.

Let's fill the database. You can do this via the command line, as show here (output not shown):

```
C:\>cd /d d:\mysql
D:\mysql>mysql -u root -p password -e 'create database otrs'
D:\mysql>mysql -u root -p password otrs < d:\otrs\install\database\OpenTRS-SQL-win32.mysql.sql
D:\mysql>mysql -u root -p -e 'GRANT ALL PRIVILEGES ON otrs.* TO otrs@localhost IDENTIFIED BY "some-pass" WITH
D:\mysql>mysqladmin -u root -p password reload
```

I recommend using *phpMyAdmin* under <http://localhost/phpMyAdmin> instead of working on the command line. You create the database right on the start screen. Now you click the tab "SQL", and enter *d:\otrs\install\database\OpenTRS-SQL-win32.mysql.sql*. Click OK.

Now commit this command:

```
GRANT ALL PRIVILEGES ON otrs.* TO otrs@localhost IDENTIFIED BY "irgendeinpasswort" WITH GRANT OPTION;
```

That's it for MySQL, which we provided with the database otrs and the user otrs@localhost.

1.3.2.4. Configuring OTRS

1.3.2.4.1. CONFIG.PM

Now you teach OTRS s.th. about you. First create your configuration file:

```
C:\>cd /d d:\otrs\Kernel
D:\otrs\bin>copy Config.pm.dist Config.pm
```

Despite your settings concerning FQDN etc., you set the following in *d:\otrs\Kernel\Config.pm*:

```
# -----#
# Sendmail
# -----#
$Self->{'SendmailModule'} = 'Kernel::System::Email::SMTP';
$Self->{'SendmailModule::Host'} = 'smtp-server.of.your.provider';
$Self->{'SendmailModule::AuthUser'} = 'smtp-username';
$Self->{'SendmailModule::AuthPassword'} = 'smtp-password';
# -----#
# directories
# -----#
$Self->{Home} = 'd:/otrs';
$Self->{CounterLog} = '<OTRS_CONFIG_Home>/var/log/TicketCounter.log';
$Self->{ArticleDir} = '<OTRS_CONFIG_Home>/var/article';
$Self->{StatsPicDir} = '<OTRS_CONFIG_Home>/var/pics/stats';
$Self->{TemplateDir} = '<OTRS_CONFIG_Home>/Kernel/Output';
$Self->{TempDir} = '<OTRS_CONFIG_Home>/var/tmp';
# -----#
# LogModule
# -----#
$Self->{LogModule} = 'Kernel::System::Log::File';
# -----#
# own config settings
#
# config settings taken from Kernel/Config/Defaults.pm #
```

```
# -----#
# $Self->{SessionUseCookie} = 0;
$Self->{SessionModule} = 'Kernel::System::AuthSession::DB';
```

A note about your FQDN: This is the called *Full Qualified Domain Name* of your PC, just like your IP adress it's unique across the universe ...ah... internet, I mean. If you're using a dial-up line to connect to the internet, your FQDN will always be a different one each time you dial anew. This is indeed ok for OTRS, but it's not nice if exim announces itself with a wrong name. It can lead to difficulties in mail delivery.

To solve this problem you can use so called *Dynamic DNS services* like <http://www.dyndns.org>. By using a client such as *DirectUpdate* (<http://www.directupdate.net>) you submit your IP adress to *dyndns.org* each times it changes and can thereby obtain a FQDN, for example *gogos-pc.home-ip.net*. You can even forward your domain <http://www.thisismydomain.com> to your home PC. If you got further questions concerning Dynamic DNS services, don't hesitate to ask(<mailto:otrs-win32@robertkehl.de?subject=Dynamic%20DNS>) - but now back to OTRS :)

1.3.3. The First Start

Now startup OTRS for the first time. A good idea is to reboot your machine before, remember: You're on Windows ;) Then call <http://localhost/otrs/index.pl> in your favourite browser.

The user name is "root@localhost", the password is "root", both without the quotes.

Important: It's a **very good** idea to **now** change your password on <http://localhost/otrs/index.pl?Action=AdminUser>

1.3.3.1. System Email Addresses & POP3

You set your system email addresses here: <http://localhost/otrs/index.pl?Action=AdminSystemAddress>. These are the addresses your customers lateron use to reach your OTRS.

You create POP3 accounts for each system email here: <http://localhost/otrs/index.pl?Action=AdminPOP3>

Your OTRS is now ready, you can start! Have a look around the interface and work on the test ticket provided.

Now write an email to one of the system email addresses. Enter this in your browser, it provides the email to OTRS. <http://localhost/obin/PostMasterPOP3.pl>. We will later automate this.

Your email will be visible in the queue *Raw* - that's here: <http://localhost/otrs/index.pl?Action=AgentQueueView>. Answer this first inquiry.

Tip: It is *NO* good idea to use one of the System Email Addresses to address the system, this will produce loops.

1.3.3.2. Setup CRON Job(s)

At last, we tell Sambar to fetch all POP3-Mail every 10 minutes and pump it into the database. Therefore you enter this in *D:\Sambar\config\schedule.ini*:

```
[cron]
# fetch emails every 10 minutes
*/10 * * * * perl d:/otrs/bin/PostMasterPOP3.pl &
```

Important: The schedule entry will be active 1-2 minutes after saving the file.

Tip: You may enter the following CRON jobs at this very moment, too. Their very purpose is described elsewhere in this manual.

```
[cron]
```

```
# start generic agent every 20 minutes
*/20 * * * * perl d:/otrs/bin/GenericAgent.pl &

# check every 60 min the pending jobs
45 * * * * perl d:/otrs/bin/PendingJobs.pl &

# Rebuild Ticket Index every day
01 01 * * * * perl d:/otrs/bin/RebuildTicketIndex.pl &

# delete every six hours old session ids
55 */6 * * * * perl d:/otrs/bin/DeleteSessionIDs.pl --expired &

# unlock every hour old locked tickets
35 * * * * perl d:/otrs/bin/UnlockTickets.pl --timeout &
```

No CRON Jobs on Apache: Please note that Apache cannot CRON. You will have to execute the CRON Jobs manually from the command line or by using Windows Task Scheduler. Use the following command for each script to prevent windows popping up at you screen while executing the script. `cmd /c start /min perl d:\otrs\bin\NameOfCRONJob.pl`

1.3.4. Done!

Congratulations, you hopefully were successful in installing OTRS on Win32!

If s.th. still is not as clear as it should be or if you are getting nothing more than error messages instead of the interface of OTRS, don't hesitate to mail `otrs-win32` at `robertkehl.de` (`mailto:otrs-win32@robertkehl.de`). Hopefully you get a ticket back :-)

I would be more enlightened if you'd send some success stories!

Surely this chapter will follow the development of OTRS(<http://otrs.org>). OTRS on Win32 can be done with Apache, too, and I will soon include a part on it.

Have fun using your OTRS on Sambar!

Chapter 2. Basics about a Trouble Ticket System

2.1. A simple example for a small Trouble Ticket System

What is a typical scenario for using OTRS?

Example: Bob is a manufacturer of VCRs and his customers often have problems to program the VCRs. So they send Bob an e-mail. Sometimes they send a second e-mail to show Bob how important their request is. They are wondering if Bob is alive and how fast he will answer. Bob is using a normal INBOX and reads his e-mails with pine, mutt or what ever e-mail client. Sometimes his two brothers Tim and Joe help him to answer the e-mails. They all use the same INBOX. Of course they have no clue that one customer wrote two e-mails and maybe Tim gives a different answer to the first e-mail than his brother Joe does for the second. So the client gets different information. Of course Bob has no client-history and no clue how much support he is offering. For the next VCR he is producing he has no feedback from his support. That is bad!

But Bob is a smart cookie so he installs OTRS. The e-mail from his customers are not anymore going to his personal INBOX but are routed to the OTRS account (normally called otrs). The OTRS account has some nifty procmail rules which pipe this e-mail messages to the system. The system answers the client a standard text which says that they received the e-mail and gives the client a Trouble Ticket Number (which is very important to trace the customers request). The client is happy because he knows that his valuable e-mail was received by Bob and his team. Anybody from Bob's team can open a webbrowser with the URL of the OTRS to have a look at the amount of received e-mails and to answer them. In case the customer Mr. Smith sent a question, Bob can answer it. Maybe Mr. Smith does not understand the question and sends a reply. But Bob is ill. Now even Tim can open this ticket and has access to the history of the ticket. He can read Bob's answer(s) and the original e-mail of Mr. Smith. Tim can answer to Mr. Smith and Mr. Smith even does not realize that he was handled by two different persons.

Of course this is just a very rough overview of the benefits of OTRS. Probably Bob receives some 100 e-mail messages a day which could be handled even without a Trouble Ticket System. But by the time you receive some 100000 or even just some 500 e-mails a day you will be happy to have a system which handles all the e-mails.

2.2. What is a trouble ticket in the OTRS?

Within the OTRS all trouble tickets are handled as normal e-mails. In case you want to attach something (e.g. a fax) it will be attached as an e-mail attachment. All tickets are stored on the harddrive in clear text format. The headers are stored in a database too. The database is used to sort the tickets and to give quick access to them. For detailed information about this mechanism have a look in the source code.

2.3. What is a ticket queue?

For native english speaking folks this might be a bit funny but for non native speakers the term QUEUE does not make any sense at all. So we use this section to describe the idea and concept.

Normally an e-mail (and as described above a trouble ticket is stored as an e-mail) is stored in an INBOX. An INBOX is a large file and every new e-mail is just appended to the end of the INBOX. The e-mail client is parsing this file and sorts it as you want it (typically by date of receiving).

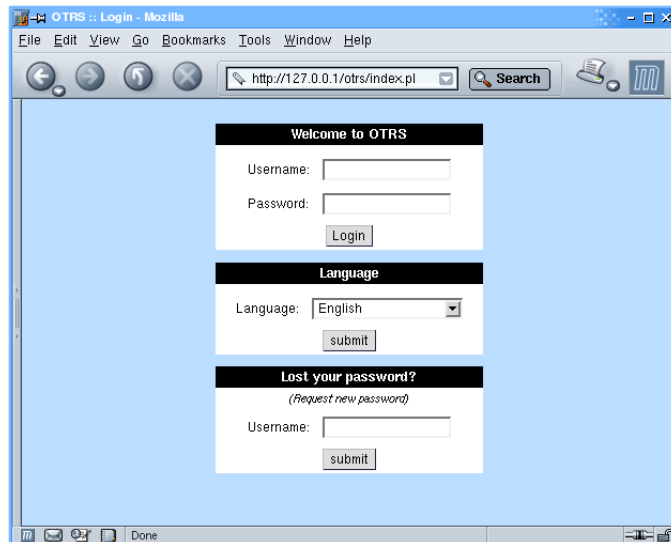
A queue is a mechanism to store many tickets within. As a user you do not know where the tickets are stored. You just know that a new ticket is e.g. in the RAW queue. A user (agent) can move a ticket from one queue to another. Why should he/she do this? You can use different queues to get more order and a better overview to your tickets. Let's assume you receive 200 e-mail messages (tickets) a day. And you have 3 teams of specialists. It doesn't make any sense to ask every specialist to read every ticket. It is a waste of time. So you have to create a fourth team which dispatches all the tickets in the INCOMING-QUEUE (or how ever you call this queue). The dispatch instance will have a quick (quick!) look at every ticket and move it then to a special queue. The 3 teams of specialists read their special queues only.

Chapter 3. First steps

3.1. Login as root and create a new account

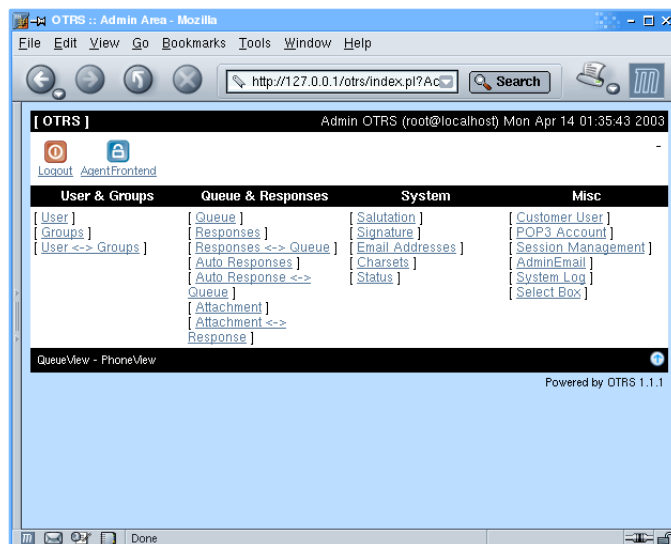
Let's presume that you have an installed OTRS system on your system and of course you do not want to waste too much time and see results quickly.

First you have to start your favourite webbrowser and have a look at <http://localhost/otrs/index.pl>

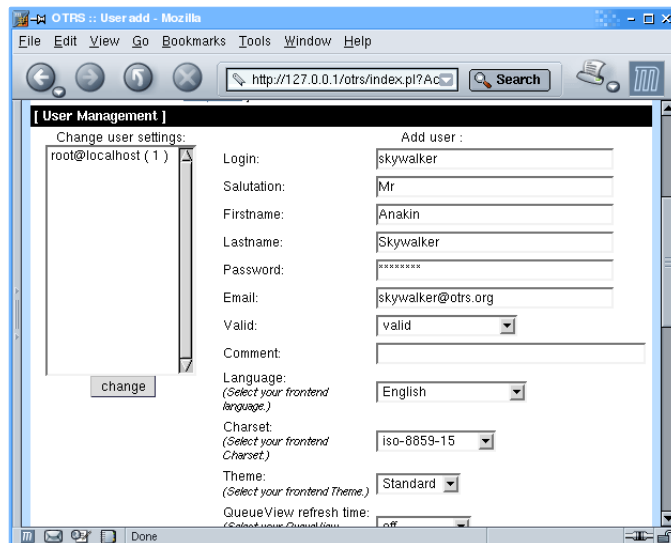


Login as root@localhost who by default has assigned the password root. Please change this in the admin part asap (of course it is totally independent from your normal linux root account).

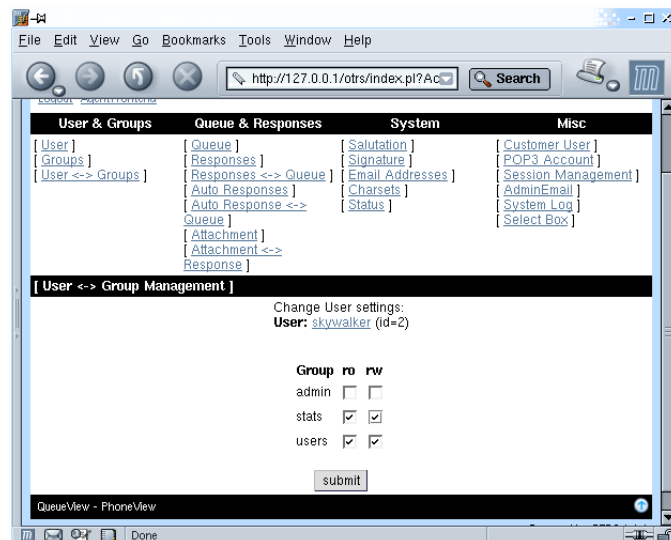
You are the root user of OTRS now. That means that you can do everything! You have the power, you are admin of the system. Normally you will not want to work as root and of course you need an account for all agents. So the first thing is to go into the admin interface.



The admin interface is the central of your power. You can create and delete users, groups, queues and all sort of useful stuff here. Browse around and give it a try. But for now we want to create a new user at once.



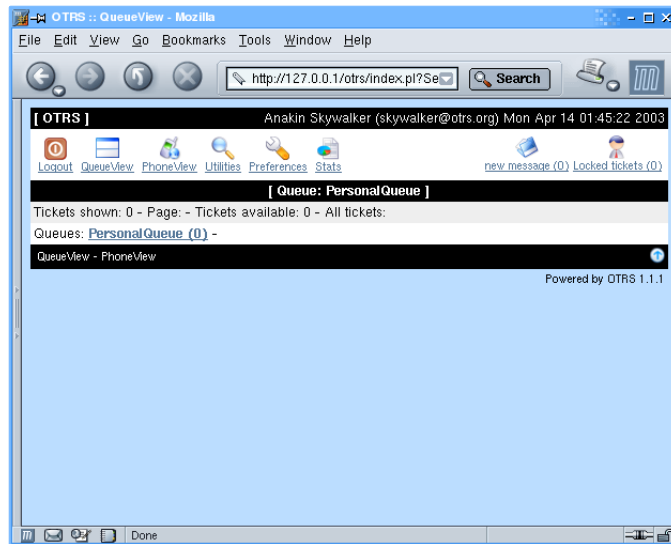
Now add the new user to the 'users' and 'stats' group or some other groups.



3.2. Login with user privilege

After you created the new user we ask you to logout and to login again as this new user. After login in he/she will see the

following screen:



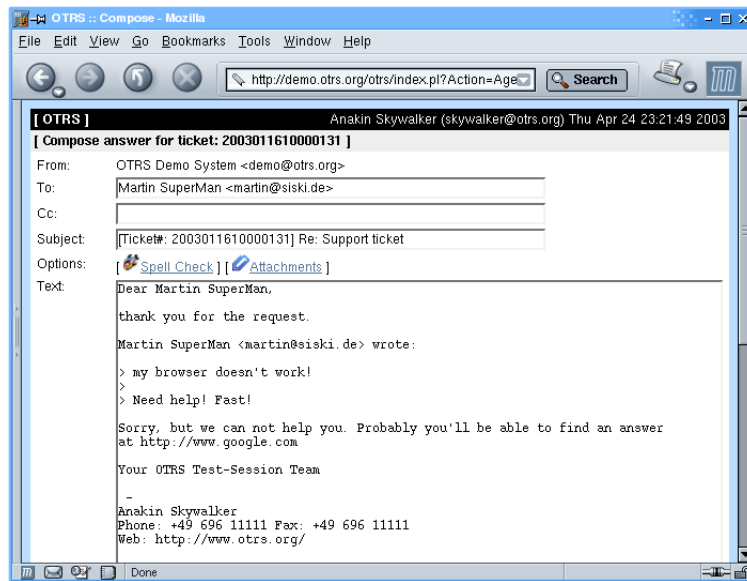
Between the second and third black bar (in this example) the user has access to the different queues. He/she can browse the queue by simply clicking on the link with the queue name.



Chapter 4. Ticket topics

4.1. Answer ticket via email

It's quite simple to answer tickets via email. Goto the QueueView or TicketZoom and click on one of the listed items under "Compose Answer (email)". A new screen will be opened, the compose screen. The cool thing is, that the compose screen includes the item-standard answer!




The screenshot shows the OTRS Compose screen in a Mozilla browser window. The address bar shows the URL <http://demo.otrs.org/otrs/index.pl?Action=Age>. The page title is "OTRS :: Compose - Mozilla". The main content area displays the "Compose answer for ticket: 2003011610000131" form. The form includes fields for "From:", "To:", "Cc:", "Subject:", and "Options:". The "Text:" field contains the following content:

Dear Martin SuperMan,
thank you for the request.
Martin SuperMan <martin@siski.de> wrote:
> my browser doesn't work!
>
> Need help! Fast!
Sorry, but we can not help you. Probably you'll be able to find an answer
at <http://www.google.com>
Your OTRS Test-Session Team
-
Anakin Skywalker
Phone: +49 696 11111 Fax: +49 696 11111
Web: <http://www.otrs.org/>

4.2. Answer ticket via phone

It's quite simple to answer tickets via phone. Just click on "phone call" (near Contact customer (phone)). A new screen will be opened. Write down the phone notes and select new ticket type (open, closed, ...).



The screenshot shows the OTRS Phone call screen in a Mozilla browser window. The address bar shows the URL <http://demo.otrs.org/otrs/index.pl?Action=Age>. The page title is "OTRS :: Phone call - Mozilla". The main content area displays the "Phone call: 2003011610000131" form. The form includes fields for "Subject:", "Options:", and "Text:". The "Text:" field contains the following content:

Customer called

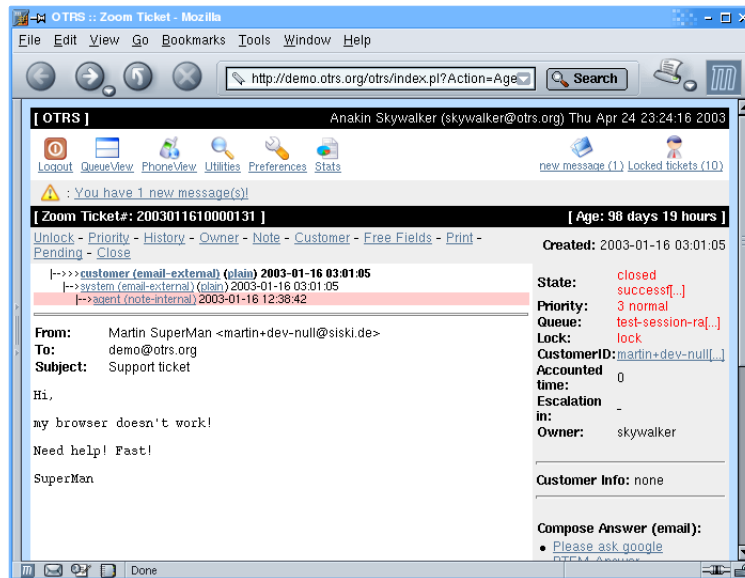
Next ticket state: closed successful

Pending Date (for pending* states): 04 - 25 - 2003 - 23 - 23

Is the ticket: Yes

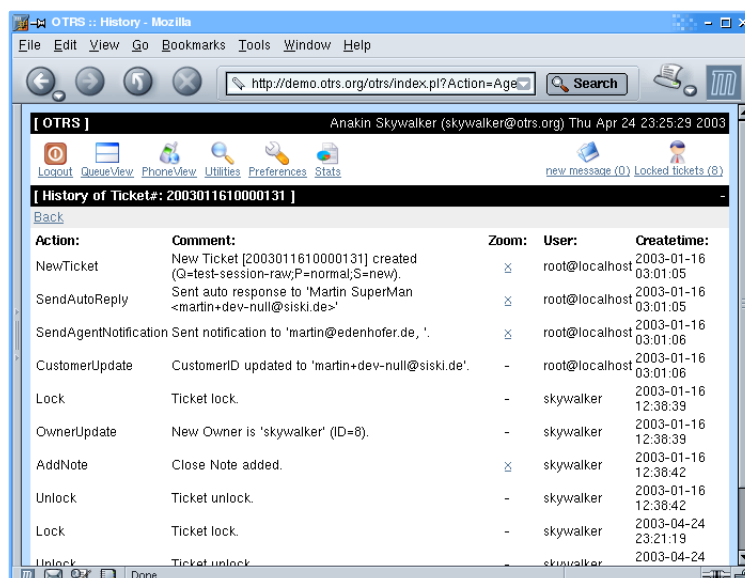
4.3. Zoom ticket

Zoom ticket is a detailed view of one ticket. The screenshot shows you the ticket data (State, Priority, the Queue of the ticket, the Lock state, Owner and the article tree. Article tree means the whole communication thread.



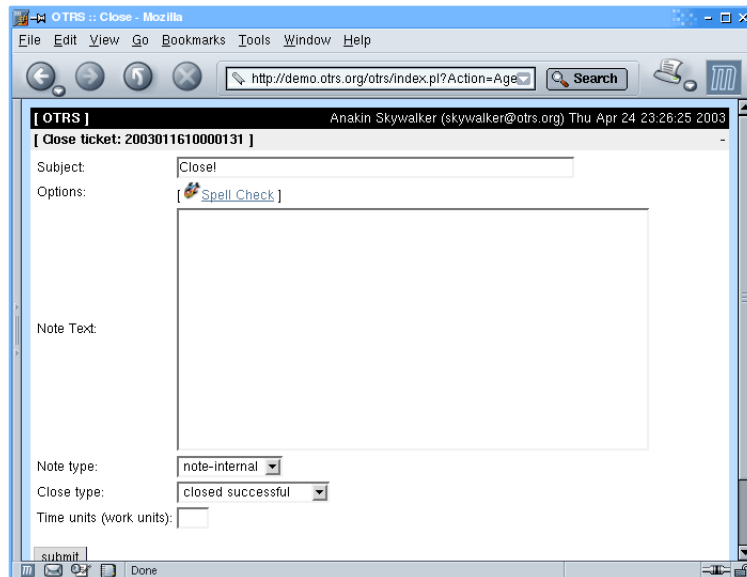
4.4. History of ticket

The ticket history shows you all actions on this ticket.



4.5. Add note to ticket

Each agent is able to add notes to tickets. Maybe he/she isn't sure to give a qualified answer. And you can select the type of note (internal, external, ...). the customer gets not notification about a new note.



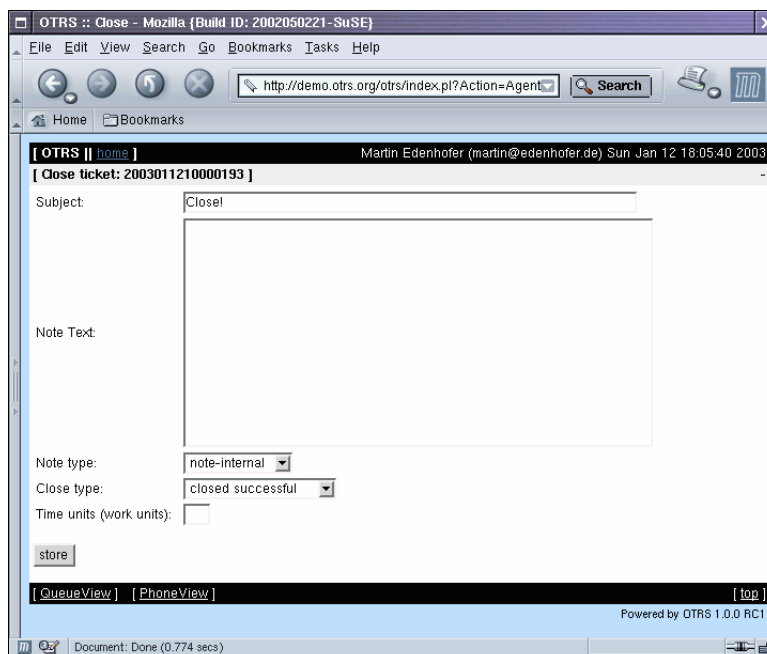
The screenshot shows a Mozilla browser window titled "OTRS :: Close - Mozilla". The address bar displays "http://demo.otrs.org/otrs/index.pl?Action=Age". The page header indicates the user is "Anakin Skywalker (skywalker@otrs.org)" and the date is "Thu Apr 24 23:26:25 2003". The main content area is titled "[Close ticket: 2003011610000131]". It contains a form with the following fields:

- Subject:** A text input field containing the text "Close!".
- Options:** A button labeled "Spell Check".
- Note Text:** A large, empty text area for adding the note.
- Note type:** A dropdown menu currently set to "note-internal".
- Close type:** A dropdown menu currently set to "closed successful".
- Time units (work units):** An empty input field.

At the bottom left of the form is a "submit" button. The browser's status bar at the bottom shows "Done".

4.6. Close ticket

Close Tickets.



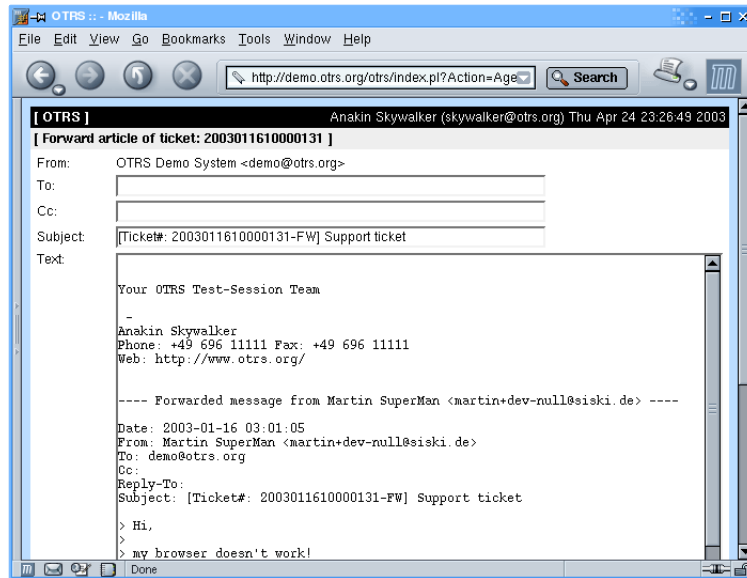
The screenshot shows a Mozilla browser window titled "OTRS :: Close - Mozilla (Build ID: 2002050221-SuSE)". The address bar displays "http://demo.otrs.org/otrs/index.pl?Action=Agent". The page header indicates the user is "Martin Edenhofer (martin@edenhofer.de)" and the date is "Sun Jan 12 18:05:40 2003". The main content area is titled "[Close ticket: 2003011210000193]". It contains a form with the following fields:

- Subject:** A text input field containing the text "Close!".
- Note Text:** A large, empty text area for adding the note.
- Note type:** A dropdown menu currently set to "note-internal".
- Close type:** A dropdown menu currently set to "closed successful".
- Time units (work units):** An empty input field.

At the bottom left of the form is a "store" button. Below the form, there are links for "[QueueView]" and "[PhoneView]", and a "[top]" link on the right. The footer of the page states "Powered by OTRS 1.0.0 RC1". The browser's status bar at the bottom shows "Document: Done (0.774 secs)".

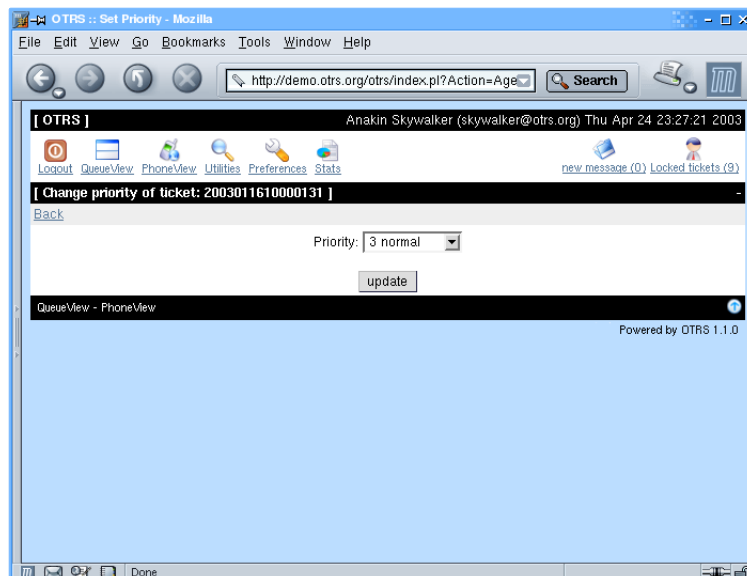
4.7. Forward ticket

Forward tickets if the email wasn't for your OTRS system.



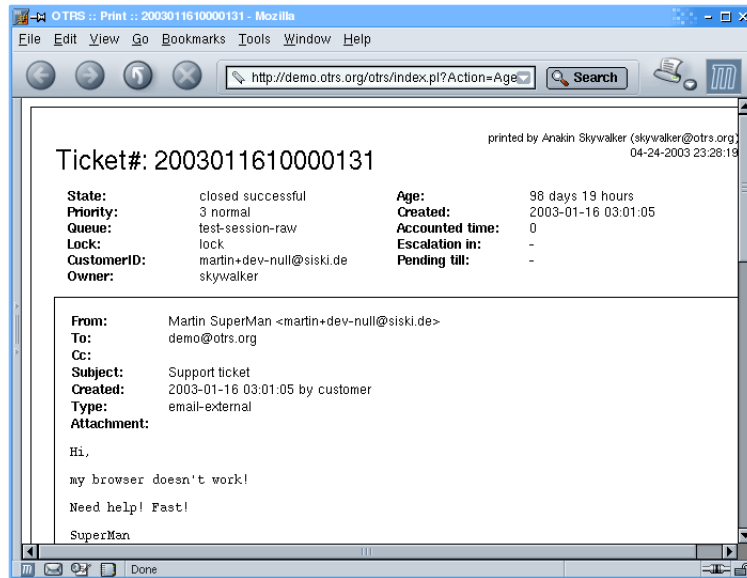
4.8. Ticket priority

Set the ticket priority if necessary.



4.9. Ticket print view

The ticket print view.



Chapter 5. System-Email-Notifications

This chapter will show you more details about the system notifications for agents and customers.

5.1. Agent-Notifications

A agent can get notifications about 'new ticket', 'ticket follow up', 'ticket in custom queue' and 'ticket lock timeout'. In order to get notifications select you wanted queues (named 'custom queues') in agent interface -- Preferences -- Select your custom queues. The notification text can be defined in Kernel/Config.pm.

Important: The agent gets just notifications about changes the 'custom queues'.

5.1.1. new ticket

Notification to the agent about new created tickets in a queue.

5.1.2. ticket follow up

Notification to the agent about a follow up for a ticket which is owned by the agent.

5.1.3. new ticket in custom queue

Notification to the agent if a ticket is moved to one of the 'custom queues'.

5.1.4. ticket lock timeout

Notification to the agent if the OTRS system unlocked a unanswered ticket.

5.2. Customer-Notifications (min 1.1)

There are three types of (auto) customer notifications to give more transparency to the customer. This notifications (if used or not) can be defined for each queue (admin area). The notification text can be defined in Kernel/Config.pm.

5.2.1. ticket state

The customer (sender) gets automatically a notification if the ticket state has changed.

5.2.2. ticket owner

The customer (sender) gets automatically a notification if the ticket owner has changed.

5.2.3. ticket queue

The customer (sender) gets automatically a notification if the ticket queue has changed.

Chapter 6. Auto-Response

This chapter will show you more details about the auto responses to customers after customer actions. Of course auto-response need to be configured via the Admin Interface and for each queue.

6.1. Reply

Used if a new ticket is created the customer (sender) gets the following auto-response.

A normal reply auto-response should contain something like that:

Thanks for your e-mail. A new ticket has been created.

You wrote:
<OTRS_CUSTOMER_EMAIL[6]>

Your e-mail will be answered by a human asap

Have fun with OTRS! :-)

Your OTRS Team

6.2. Follow up

Used if the customer wrote a follow up via email or Customer Interface.

A normal follow up auto-response should contain something like that:

Thanks for your follow up e-mail

You wrote:
<OTRS_CUSTOMER_EMAIL[6]>

Your e-mail will be answered by a human asap.

Have fun with OTRS!

Your OTRS Team

6.3. Reject

Used if the customer wrote a follow up and the follow up is rejected (because the ticket is closed and not possible to reopen).

A normal reject auto-response should contain something like that:

Thanks for your follow up e-mail

You wrote:
<OTRS_CUSTOMER_EMAIL[6]>

Important: Sorry but we can't process your follow up because your ticket is closed. Please create a new ticket.

Have fun with OTRS!

Your OTRS Team

6.4. Closed -> new Ticket

Used if the customer wrote a follow up and the follow up is rejected and a new ticket is created (because the ticket is closed and not possible to reopen).

A normal closed -> new ticke auto-response should contain something like that:

Thanks for your follow up e-mail

You wrote:

<OTRS_CUSTOMER_EMAIL[6]>

Important: Sorry but we can't process your follow up because your ticket is closed. A new ticket has been created automatically.

Have fun with OTRS!

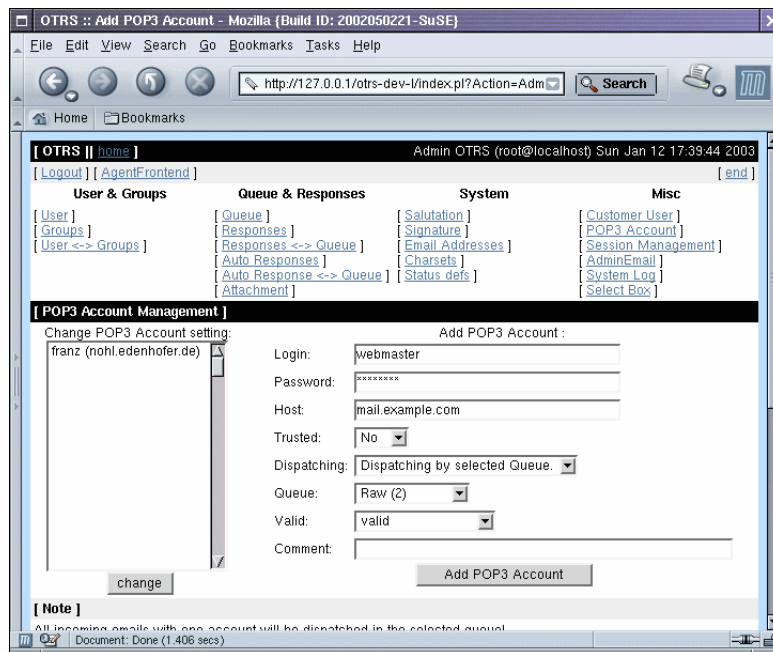
Your OTRS Team

Chapter 7. Receiving emails

7.1. Via POP3 account

OTRS is able to receive emails from POP3 accounts.

Configure your POP3 accounts via the admin interface (POP3 Account).



Execute bin/PostMasterPOP3.pl and all emails will be fetched to your OTRS system.

There is also an example cronjob (var/cron/postmaster_pop3.dist) which execute your bin/PostMasterPOP3.pl every 10 minutes (see also chapter cronjobs).

7.2. Via command line program and e. g. procmail

OTRS is able to receive emails via a command line programm (bin/PostMaster.pl).

That means emails will be shown in your OTRS system if the MDA (mail delivery agent, e. g. procmail) pipes the emails into bin/PostMaster.pl.

To test the bin/PostMaster.pl on your command line (without MDA) use:

```
shell:~ # cat /opt/otrs/doc/test-email-1.box | /opt/otrs/bin/PostMaster.pl
shell:~ #
```

If the email is shown in the QueueView then your setup works fine.

Procmail is a very common e-mail filter in the Linux enviroment. It will be probably installed on your system. If not have a look at the *procmail homepage*(<http://www.procmail.org/>).

To configure procmail for that (requires a procmail configured MTA (e. g. sendmail, postfix, exim or qmail)) use the ~otrs/.procmailrc and modify/add the following.

```
SYS_HOME=$HOME
PATH=/bin:/usr/bin:/usr/local/bin
# --
# Pipe all email into the PostMaster process.
# --
:0 :
| $SYS_HOME/bin/PostMaster.pl
```

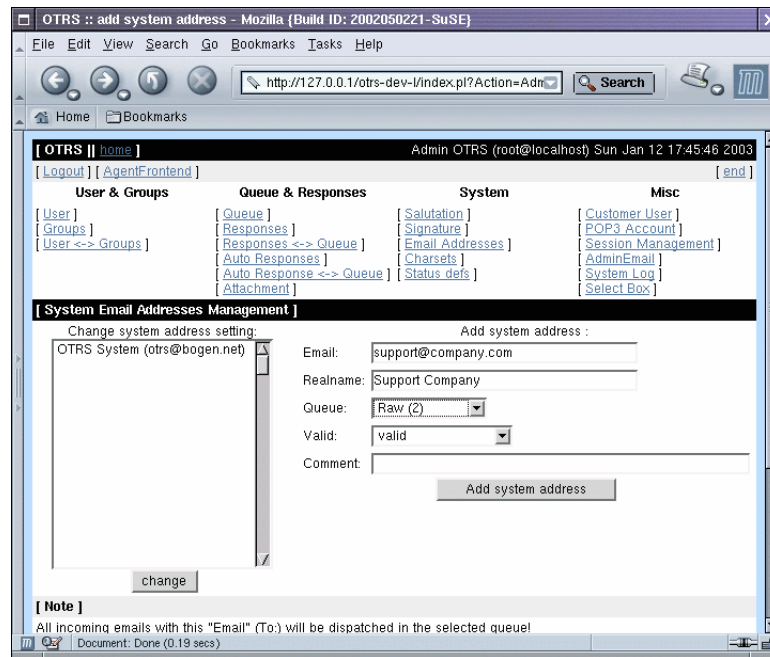
All emails sent to the local otrs user will be piped into bin/PostMaster.pl and then shown in your QueueView.

7.2.1. Queue dispatching with OTRS

OTRS is able to dispatch incoming emails via "To" and "Cc" address. Configurable via admin interface.

Example

Add a new system email address (AdminArea --> System Email Addresses).



In this case, all incoming emails (with To: or Cc: support@company.com) will be dispatched to the TEST1 queue.

7.2.2. Queue dispatching with procmail (for more complex dispatching)

The X-OTRS-Queue Mail-Header

The X-OTRS-Queue Mail-Header is parsed by OTRS and OTRS will pipe these e-mails direct in this queue. Procmail and fromail can be used to create a powerfull dispatcher tool.

Examples

The following examples are copied from the procmailex man-page. Feel free to have a look into it (actually it is quite a good idea). Of course we changed the wording a bit (to fit it into the queueing idea).

Sort out all mail coming from the scuba-dive mailing list into the scuba queue.

```
:0 fhw :
* ^TOscuba
| formail -I "X-OTRS-Queue: scuba"
```

Forward all mail from peter about compilers into the william queue.

```
:0 fhw :
* ^From.*peter
* ^Subject:.*compilers
| formail -I "X-OTRS-Queue: william"
```

And here a last example, the whole .procmailrc.

Example 7-1. .procmailrc

```
# --
# .procmailrc - procmailrc of the OTRS user
# Copyright (C) 2001-2002 Martin Edenhofer (martin+code at otrs.org)
# --
# $Id: receiving_email.sgml,v 1.4 2003/02/03 21:12:19 martin Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY. For details, see
# the enclosed file COPYING for license information (GPL). If you
# did not receive this file, see http://www.gnu.org/licenses/gpl.txt.
# --

SYS_HOME=$HOME

PATH=/bin:/usr/bin:/usr/local/bin
MONTHFOLDER='date +%Y-%m'
YEARFOLDER='date +%Y'
LOGFILE=$SYS_HOME/var/log/procmail-$MONTHFOLDER.log
VERBOSE=on

# --
# Remove all X-OTRS Header (allow this only for trusted email)
# e. g. from *@example.com
# --
:0 fhw :
* !^From.*@example.com
| grep -vi '^X-OTRS-'

# --
# Examples for queue presorting.
# --

:0 fhw :
* ^List-Id:.*OpenAntiVirus
| formail -I "X-OTRS-Queue: OpenAntiVirus"

:0 fhw :
* ^Sender:.*example.com
| formail -I "X-OTRS-Queue: example"

:0 fhw :
* TO:.*BUGTRAQ
| formail -I "X-OTRS-Queue: BUGTRAQ"

# --
# Backup of all incoming emails.
# It's always better to have a backup of all incoming emails!
# --
:0 c :
$SYS_HOME/var/INBOX.Backup.$MONTHFOLDER

# --
# Pipe all email into the PostMaster process.
# --
:0 :
| $SYS_HOME/bin/PostMaster.pl

# --
# spool all the rest (which the PostMaster.pl can't process!)
# If the database is down or the PostMaster.pl exit was not '0!'
# --
:0 :
$SYS_HOME/var/spool/

# --
# end of .procmailrc
# --
```

Please have a look into the procmailex man-page for more examples.

7.2.3. Example with procmail and a webform

This is an example for a webform to generate an email for OTRS. You will find this perl script in `$OTRS_HOME/scripts/webform.pl`

You have a Topic, From, Email, Subject and Message field.

Change the config settings for the webform:

```
# --
# web form options
# --
my $Ident = 'ahfiw2Fw32r230dddl2foeo3r';
# sendmail location and options
my $Sendmail = '/usr/sbin/sendmail -t -i -f ';
# email where the emails of the form will send to
my $OTRSEmail = 'otrs-system@example.com';
# topics and dest. queues
my %Topics = (
    # topic => OTRS queue
    'Info' => 'info',
    'Support' => 'support',
    'Bugs' => 'bugs',
    'Sales' => 'sales',
    'Billing' => 'billing',
    'Webmaster' => 'webmaster',
);
```

Take care, that your used \$OTRSEmail and the used OTRS queues exists in your OTRS system.

Now, change the OTRS .procmailrc from:

```
# --
# Remove all X-OTRS Header (allow this only for trusted email)
# e. g. from *@example.com
# --
:0 fhw :
* !^From.*@example.com
| grep -vi '^X-OTRS-'

to:

# --
# Remove all X-OTRS Header (allow this only for trusted email)
# just not emails with "X-OTRS-Ident: ahfiw2Fw32r230dddl2foeo3r" header!
# --
:0 fhw :
* !^X-OTRS-Ident: ahfiw2Fw32r230dddl2foeo3r
| grep -vi '^X-OTRS-'
```

If a email is generated by the webform.pl and sent to the \$OTRSEmail it will be dispatched to the topic=>queue.

7.2.4. Fetching emails via POP3 or IMAP and fetchmail

In order to get e-mails from your mail server via a POP3 or IMAP mailbox to the **OTRS machine/local otrs account and to procmail** use fetchmail(<http://www.tuxedo.org/~esr/fetchmail/>). Note: A working SMTP configuration on the OTRS machine is a condition.

Example 7-2. .fetchmailrc

```
#poll (mailserver) protocol POP3 user (user) password (password) is (localuser)
poll mail.example.com protocol POP3 user joe password mama is otrs
```

Don't forget to set the .fetchmailrc to 710 ("chmod 710 .fetchmailrc")!

So if "fetchmail -a" is executed (maybe via cron), all e-mails will be forwarded to the local otrs account.

Chapter 8. Sending emails

8.1. Via Sendmail (default)

OTRS is able to send out emails via Sendmail (e. g. Sendmail(<http://www.sendmail.org/>), Postfix(<http://www.postfix.org/>), Qmail(<http://www.qmail.org>) or Exim(<http://www.exim.org>)). The default configuration to use Sendmail should work out of the box.

There are the following config options for Kernel/Config.pm:

```
# SendmailModule
# (Where is sendmail located and some options.
# See 'man sendmail' for details.)
$Self->{'SendmailModule'} = 'Kernel::System::Email::Sendmail';
$Self->{'SendmailModule::CMD'} = '/usr/sbin/sendmail -t -i -f ';
```

8.2. Via SMTP relay/smarthost (min. OTRS 1.1)

OTRS is able to send out emails via SMTP (Simple Mail Transfer Protocol / RFC 821(<http://www.ietf.org/rfc/rfc821.txt>)). Mostly used on non unix platforms (e. g. win32).

There are the following config options for Kernel/Config.pm:

```
# SendmailModule
$Self->{'SendmailModule'} = 'Kernel::System::Email::SMTP';
$Self->{'SendmailModule::Host'} = 'mail.example.com';
$Self->{'SendmailModule::AuthUser'} = "";
$Self->{'SendmailModule::AuthPassword'} = "";
```

Chapter 9. User

This chapter will show you more details about the system users (agents).

9.1. How it works

You need system users (agents) to deal with all the tickets.

9.1.1. Admin-Interface

Edit your system user via the Admin-Interface.

9.2. User Backend

There is no existing user backend. At the moment the OTRS user need to be in the otrs database.

User backends (for DB and LDAP) are planned for OTRS 2.0.

9.3. User Auth Backend

9.3.1. Database (default)

The default user auth backend is the otrs database.

```
[Kernel/Config.pm]
# This is the auth. module against the otrs db
$Self->{'AuthModule'} = 'Kernel::System::Auth::DB';
[...]
```

9.3.2. LDAP

If there is a LDAP tree available with your users then you can use the user LDAP auth backend. This module is just read only (means it can't write to your LDAP tree - this should just be possible for your tree manager) so you can't create or update user via the Admin-Interface.

```
[Kernel/Config.pm]
# This is an example configuration for an LDAP auth. backend.
# (take care that Net::LDAP is installed!)
$Self->{'AuthModule'} = 'Kernel::System::Auth::LDAP';
$Self->{'AuthModule::LDAP::Host'} = 'ldap.example.com';
$Self->{'AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';
$Self->{'AuthModule::LDAP::UID'} = 'uid';

# Check if the user is allowed to auth in a posixGroup
# (e. g. user needs to be in a group xyz to use otrs)
# $Self->{'AuthModule::LDAP::GroupDN'} = 'cn=otrsallow,ou=posixGroups,dc=example,dc=com';
# $Self->{'AuthModule::LDAP::AccessAttr'} = 'memberUid';

# The following is valid but would only be necessary if the
# anonymous user do NOT have permission to read from the LDAP tree
$Self->{'AuthModule::LDAP::SearchUserDN'} = '';
$Self->{'AuthModule::LDAP::SearchUserPw'} = '';
[...]
```

LDAP entries shall conform to the PosixAccount and inetOrgPerson schema. An example entry might look like:

```
dn: uid=lester,ou=user,dc=example,dc=com
objectClass: top
objectClass: account
```

```
objectClass: posixAccount
objectClass: organizationalPerson
objectClass: inetOrgPerson
objectClass: officePerson
uid: lester
cn: Lester Adamas
userPassword: {crypt}X5/DBrWPOQQaI
gecos: Lester
loginShell: /bin/csh
uidNumber: 10
gidNumber: 10
homeDirectory: /home/lester
sn: Adams
givenName: Lester
mail: lester@example.com
preferredLanguage: fr
comment: technical support
```

Please note, that in version 1.x you have to create a db entry for the ldap user. This is done automatically on first login in versions < 1.1

Chapter 10. Customer (min. OTRS 1.1)

OTRS is able to deal with various pieces of customer information (i.e. login, email, phone). This information can be shown in the Agent-Interface and can be used for the Customer-Interface (users and auth).

10.1. How it works

The used/shown customer information is freely configurable (see backend), but there are three required options: UserLogin, UserEmail and UserCustomerID.

10.1.1. Agent-Interface

If you want to show the customer user information (e. g. company, name, email, ...) in your Agent-Interface use the following config options.

```
[Kernel/Config.pm]
# ShowCustomerInfo*
# (show customer user info on Phone, Zoom and Queue view)
$Self->{ShowCustomerInfoPhone} = 1;
$Self->{ShowCustomerInfoZoom} = 1;
$Self->{ShowCustomerInfoQueue} = 0;
[...]
```

10.1.2. Admin-Interface

Edit your customer user information via the Admin-Interface (<http://yourhost/otrs/index.pl?Action=AdminCustomerUser>).

10.1.3. Customer-Interface

The customer information for the Customer-Interface (<http://yourhost/otrs/customer.pl>) is used for the login and auth.

10.2. Customer User Backend

There are two existing customer user backends, DB and LDAP. Of course if you have an existing customer repository (e. g. SAP, ...) it's possible to write an own backend.

10.2.1. Database (default)

This is an example for a database backend.

```
[Kernel/Config.pm]
# CustomerUser
# (customer user database backend and settings)
$Self->{CustomerUser} = {
    Module => 'Kernel::System::CustomerUser::DB',
    Params => {
        Table => 'customer_user',
    },
    # customer uniq id
    CustomerKey => 'login',
    # customer #
    CustomerID => 'customer_id',
    CustomerValid => 'valid_id',
    CustomerUserListFields => ['login', 'first_name', 'last_name', 'email'],
    CustomerUserSearchFields => ['login', 'last_name', 'customer_id'],
    CustomerUserNameFields => ['salutation', 'first_name', 'last_name'],
    Map => [
        # note: Login, Email and CustomerID needed!
```

```

# var, frontend, storage, shown, required, storage-type, http-link
[ 'UserSalutation', 'Salutation', 'salutation', 1, 0, 'var' ],
[ 'UserFirstname', 'Firstname', 'first_name', 1, 1, 'var' ],
[ 'UserLastname', 'Lastname', 'last_name', 1, 1, 'var' ],
[ 'UserLogin', 'Login', 'login', 1, 1, 'var' ],
[ 'UserPassword', 'Password', 'pw', 0, 1, 'var' ],
[ 'UserEmail', 'Email', 'email', 0, 1, 'var' ],
# [ 'UserEmail', 'Email', 'email', 1, 1, 'var', '$Env{"CGIHandle"}?Action=\
AgentCompose&ResponseID=1&TicketID=$Data{"TicketID"}&ArticleID=$Data{"ArticleID"}' ],
[ 'UserCustomerID', 'CustomerID', 'customer_id', 0, 1, 'var' ],
[ 'UserComment', 'Comment', 'comment', 1, 0, 'var' ],
[ 'ValidID', 'Valid', 'valid_id', 0, 1, 'int' ],
],
};
[...]
```

If you want to customize your customer user information, change (add) you table columns e. g.

```
ALTER TABLE customer_user ADD phone VARCHAR (250);
```

And add your new column to your MAP array like:

```

# var, frontend, storage, shown, required, storage-type, http-link
[ 'UserPhone', 'Phone', 'phone', 1, 0, 'var' ],
```

Of course you will be able to maintain all this customer user information via the Admin-Interface.

10.2.2. LDAP

If you have an existing LDAP tree with your customer users then you will be able to use this LDAP tree in your OTRS system.

This is an example for a LDAP backend.

```

[Kernel/Config.pm]
# CustomerUser
# (customer user ldap backend and settings)
$Self->{CustomerUser} = {
  Module => 'Kernel::System::CustomerUser::LDAP',
  Params => {
    # ldap host
    Host => 'bay.csuhayward.edu',
    # ldap base dn
    BaseDN => 'ou=seas,o=csuh',
    # search scope (one|sub)
    SSCOPE => 'sub',
    # The following is valid but would only be necessary if the
    # anonymous user does NOT have permission to read from the LDAP tree
    UserDN => "",
    UserPw => "",
  },
  # customer uniq id
  CustomerKey => 'uid',
  # customer #
  CustomerID => 'mail',
  CustomerUserListFields => ['uid', 'cn', 'mail'],
  CustomerUserSearchFields => ['uid', 'cn', 'mail'],
  CustomerUserNameFields => ['givenname', 'sn'],
  Map => [
    # note: Login, Email and CustomerID needed!
    # var, frontend, storage, shown, required, storage-type
    [ 'UserSalutation', 'Title', 'title', 1, 0, 'var' ],
    [ 'UserFirstname', 'Firstname', 'givenname', 1, 1, 'var' ],
    [ 'UserLastname', 'Lastname', 'sn', 1, 1, 'var' ],
    [ 'UserLogin', 'Login', 'uid', 1, 1, 'var' ],
    [ 'UserEmail', 'Email', 'mail', 1, 1, 'var' ],
    [ 'UserCustomerID', 'CustomerID', 'mail', 0, 1, 'var' ],
    [ 'UserPhone', 'Phone', 'telephonenumber', 1, 0, 'var' ],
    [ 'UserAddress', 'Address', 'postaladdress', 1, 0, 'var' ],
    [ 'UserComment', 'Comment', 'description', 1, 0, 'var' ],
```

```
    ],
  };
  [...]
```

If you want to customize your customer user information, add your new item (or remove it if not needed) to your MAP array like:

```
# var, frontend, storage, shown, required, storage-type, http-link
[ 'UserOrganisation', 'Organisation', 'ou', 1, 0, 'var' ],
```

10.3. Customer Auth Backend

10.3.1. Database (default)

The default customer auth backend is the otrs database with your customer user which are created via the Admin-Interface or Customer-Interface (Create Account).

```
[Kernel/Config.pm]
# This is the auth. module against the otrs db
$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::DB';
[...]
```

10.3.2. LDAP

If there is a LDAP tree available with your customer users then you can use the customer LDAP auth backend. This module is just read only (means it can't write to your LDAP tree - this should just be possible for your tree manager) so you can't create customer user via the Admin- or Customer-Interface.

```
[Kernel/Config.pm]
# This is an example configuration for an LDAP auth. backend.
# (take care that Net::LDAP is installed!)
$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::LDAP';
$Self->{'Customer::AuthModule::LDAP::Host'} = 'ldap.example.com';
$Self->{'Customer::AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';
$Self->{'Customer::AuthModule::LDAP::UID'} = 'uid';

# Check if the user is allowed to auth in a posixGroup
# (e. g. user needs to be in a group xyz to use otrs)
# $Self->{'Customer::AuthModule::LDAP::GroupDN'} = 'cn=otrsallow,ou=posixGroups,dc=example,dc=com';
# $Self->{'Customer::AuthModule::LDAP::AccessAttr'} = 'memberUid';

# The following is valid but would only be necessary if the
# anonymous user do NOT have permission to read from the LDAP tree
$Self->{'Customer::AuthModule::LDAP::SearchUserDN'} = "";
$Self->{'Customer::AuthModule::LDAP::SearchUserPw'} = "";
[...]
```

10.4. Customize Customer Self Registration

It's possible to customize the self registration of new customers via the customer.pl. So you can add more optional or required fields (like address, location, ...).

In our example we want to add a required "address" field.

10.4.1. Template

Edit your "Kernel/Output/HTML/Standard/CustomerCreateAccount.dtl" and add your wanted fields. We want an "address" field, so add:

```
[...]
<tr>
  <td>${Text{"Address"}}: </td>
  <td> <input type="text" name="Address" value="${Data{"UserAddress"}}" size="35" maxlength="50"></td>
</tr>
[...]
```

10.4.2. Customer Map

You need also to add an "address" entry to you customer source map so add to your Kernel/Config.pm the "CustomerUser" (customer source map) from Kernel/Config/Defaults.pm and add the "address" option. Then the "CustomerUser" config option should look like that:

```
[Kernel/Config.pm]
$Self->{CustomerUser} = {
  Module => 'Kernel::System::CustomerUser::DB',
  Params => {
    Table => 'customer_user',
  },
  Map => [
    # note: Login, Email and CustomerID needed!
    # var, frontend, storage, shown, required, storage-type, link
    [ 'UserSalutation', 'Salutation', 'salutation', 1, 0, 'var' ],
    [ 'UserFirstname', 'Firstname', 'first_name', 1, 1, 'var' ],
    [ 'UserLastname', 'Lastname', 'last_name', 1, 1, 'var' ],
    # our new option
    [ 'UserAddress', 'Address', 'address', 1, 1, 'var' ],
    [ 'UserLogin', 'Login', 'login', 1, 1, 'var' ],
    [ 'UserPassword', 'Password', 'pw', 0, 1, 'var' ],
    [ 'UserEmail', 'Email', 'email', 1, 1, 'var' ],
    [ 'UserCustomerID', 'CustomerID', 'customer_id', 1, 1, 'var' ],
    [ 'UserComment', 'Phone', 'comment', 1, 0, 'var' ],
    [ 'ValidID', 'Valid', 'valid_id', 0, 1, 'int' ],
  ],
  Key => 'login',
  CustomerID => 'customer_id',
};
[...]
```

10.4.3. Modify Customer Table

We also need to add a new column to our "customer_user" table.

```
mysql> ALTER TABLE customer_user ADD address VARCHAR (200);
Query OK, 0 rows affected (0.21 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

That's all, restart your webserver (if you use mod_perl), go to your <http://otrs.example.com/otrs/customer.pl> page and test it.

Chapter 11. Permissions

This chapter will show you more details about OTRS permission.

11.1. Groups - Queue

If you want to handle different groups of users with different permissions (e. g. only read ticket in a queue or one group should not have ro/rw access to a queue) you need to:

- a) create otrs-groups
- b) put the groups of users into your otrs-groups
(you have a ro/rw option)
- c) split/sort your otrs-queue to the otrs-groups

There is a config option if 'ro' queues should be shown in QueueView.

```
Kernel/Config.pm
[...]
    # QueueViewAllPossibleTickets
    # (show all ro and rw queues - not just rw queues)
    $Self->{QueueViewAllPossibleTickets} = 0;
[...]
```

11.2. Agent - Admin Interface

It's possible to grant permissions to a module in Kernel/Modules/*.pm. For example a group of users should be able to edit responses (modular default answers). Normally you need to be in the 'admin' group to do so.

How it works:

- a) create a new otrs-group named 'responses'
- b) put the group of users rw into this otrs-group
- c) add to Kernel/Config.pm

```
[...]
$Self->{'Module::Permission'}->{'AdminResponse'} = 'responses';
[...]
```
- d) add a link to the Agent Interface e. g. Kernel/Output/HTML/*/AgentNavigationBar.dtl

```
[...]
<a href="$Env{'Baselink'}Action=AdminResponse">Response</a>
[...]
```
- e) restart the webserver if you use mod_perl

After this all users in group 'responses' are able to access the AdminResponse module and to edit the responses.

Chapter 12. Config File

OTRS has many config options. There are two config files. Kernel/Config.pm and Kernel/Config/Defaults.pm.

Kernel/Config/Defaults.pm is the default config file which should not be changed. You will find all possible config options in this file. Is quite simple to read.

Kernel/Config.pm.dist is the example file for Kernel/Config.pm (custom config file) which must be copied (cp Kernel/Config.pm.dist Kernel/Config.pm).

How it works! The Kernel/Config/Defaults.pm (default config settings) is loaded at first, then the Kernel/Config.pm (custom config file). So if you want to change the default config settings, copy (copy and paste) the needed options from Kernel/Config/Defaults.pm into Kernel/Config.pm and change the values like you want.

Now we will describe some config features in this chapter (of course not all!).

12.1. TicketHook

You can configure the value of the TicketHook which is the first part of the subject of each e-mail (e.g. [MyTicket: 007]).

Example 12-1. Kernel/Config.pm - TicketHook

```
[...]
# TicketHook
# (To set the Ticket identifier. Some people want to
# set this to e. g. 'Call#', 'MyTicket#' or 'Ticket#'.)
$Self->{TicketHook} = 'Ticket#',
[...]
```

Note: Don't use 'TN', because MS Outlook is replacing "TN: 54968797" with "RE: 54968797" and then you will run in trouble.

12.2. FQDN

The OTRS full qualified domain name. Will be used for email messages id's.

Example 12-2. Kernel/Config.pm - FQDN

```
[...]
# FQDN
# (Full qualified domain name of your system.)
$Self->{FQDN} = 'yourhost.example.com';
[...]
```

12.3. Log

The OTRS log method.

Example 12-3. Kernel/Config.pm - LogModule

```
[...]
# -----#
# log settings                                     #
# -----#

# LogModule
# (log backend module)
$Self->{LogModule} = 'Kernel::System::Log::SysLog';
#   $Self->{LogModule} = 'Kernel::System::Log::File';

# param for LogModule Kernel::System::Log::File (required!)
#   $Self->{'LogModule::LogFile'} = '/tmp/otrs.log';
[...]
```

12.4. Check Email Options

There are two options to check the used emails in the OTRS frontend.

Example 12-4. Kernel/Config.pm - CheckEmailAddresses

```
[...]
# CheckEmailAddresses
# (Check syntax of used email addresses)
$Self->{CheckEmailAddresses} = 1;
[...]
```

Example 12-5. Kernel/Config.pm - CheckMXRecord

```
[...]
# CheckMXRecord
# (Check mx records of used email addresses)
$Self->{CheckMXRecord} = 1;
[...]
```

Disable this if you use OTRS in a network without external DNS lookup. Or you can't send emails via the agent interface.

12.5. Max. Postmaster email

OTRS is able to send automatically email replies. So it's possible to create an email loop (of course we check bulk headers but other strange systems may not!).

Example 12-6. Kernel/Config.pm - PostmasterMaxEmails

```
[...]
# PostmasterMaxEmails
# (Max post master daemon email to own email-address a day.
# Loop-Protection!) [default: 40]
$Self->{PostmasterMaxEmails} = 40;
[...]
```

12.6. Ticket number format

You can choose between four different OTRS formats. AutoIncrement is default.

Example 12-7. Kernel/Config.pm - TicketNumberGenerator

```
[...]
# TicketNumberGenerator
#
# Kernel::System::Ticket::Number::AutoIncrement (default) --> auto increment
#   ticket numbers "SystemID.Counter" like 1010138 and 1010139.
#
# Kernel::System::Ticket::Number::Date --> ticket numbers with date
#   "Year.Month.Day.SystemID.Counter" like 200206231010138 and 200206231010139.
#
# Kernel::System::Ticket::Number::DateChecksum --> ticket numbers with date and
#   check sum the counter will be rotated daily (my favorite)
#   "Year.Month.Day.SystemID.Counter.CheckSum" like 2002070110101520 and 2002070110101535.
#
# Kernel::System::Ticket::Number::Random -->
#   random ticket numbers "SystemID.Random" like 100057866352 and 103745394596.

$Self->{TicketNumberGenerator} = 'Kernel::System::Ticket::Number::AutoIncrement';
[...]
```

You can also create a new (your own) module. The module needs just two functions (the whole abstraction). "CreateTicketNr()" to generate a new ticket number and "GetTNByString()" to get a ticket number by string.

A nice example is the `Kernel::System::Ticket::Number::Random`(<http://otrs.org/cgi-bin/cvsweb.cgi/otrs/Kernel/System/Ticket/Number/Random.pm>) module.

12.7. Database settings

Example 12-8. Kernel/Config.pm - Database settings

```
[...]
# DatabaseHost
# (The database host.)
$Self->{DatabaseHost} = 'localhost';

# Database
# (The database name.)
$Self->{Database} = 'otrs';

# DatabaseUser
# (The database user.)
$Self->{DatabaseUser} = 'otrs';

# DatabasePw
# (The password of database user.)
$Self->{DatabasePw} = 'some-pass';

# DatabaseDSN
# (The database DSN for MySQL ==> more: "man DBD::mysql")
$Self->{DatabaseDSN} = "DBI:mysql:database=$Self->{Database};host=$Self->{DatabaseHost}";

# (The database DSN for PostgreSQL ==> more: "man DBD::Pg")
$Self->{DatabaseDSN} = "DBI:Pg:dbname=$Self->{Database}";
[...]
```

12.8. ASP (Application Service Provider) options

Should the agent be able to move tickets into queue which it isn't member of the group (Helpfully if you use OTRS in ASP (Application Service Provider) mode).

Example 12-9. Kernel/Config.pm - MoveIntoAllQueues

```
[...]
# MoveIntoAllQueues
# (Possible to move in all queue? Not only queue which
# the own groups) [1|0]
$Self->{MoveIntoAllQueues} = 1;
[...]
```

Should the agent be able to change the ticket owner to everyone? (Helpfully if you use OTRS in ASP (Application Service Provider) mode).

Example 12-10. Kernel/Config.pm - ChangeOwnerToEveryone

```
[...]
# ChangeOwnerToEveryone -> useful for ASP
# (Possible to change owner of ticket ot everyone) [0|1]
$Self->{ChangeOwnerToEveryone} = 0;
[...]
```

12.9. Custom Queue

Some people want to set the "Custom Queue" name (The queue of your favorite queues).

Example 12-11. Kernel/Config.pm - Custom Queue

```
[...]
# CustomQueue
# (The name of custom queue.)
$Self->{CustomQueue} = 'PersonalQueue';
[...]
```

12.10. Agent notification

If there is a new ticket or a follow up for a ticket, OTRS will inform the agent(s).

Example 12-12. Kernel/Config.pm - Agent notification

```
[...]
# -----#
# notification stuff                                #
# -----#
# notification sender
$Self->{NotificationSenderName} = 'OTRS Notification Master';
$Self->{NotificationSenderEmail} = 'otrs@' . $Self->{FQDN};

# new ticket
$Self->{NotificationSubjectNewTicket} = 'New ticket notification! (<OTRS_CUSTOMER_SUBJECT[10]>)';
$Self->{NotificationBodyNewTicket} = "
Hi,

there is a new ticket!

<snip>
<OTRS_CUSTOMER_EMAIL[6]>
<snip>

http://$Self->{FQDN}/otrs/index.pl?Action=AgentZoom&TicketID=<OTRS_TICKET_ID>

Your OTRS Notification Master

";

# follow up
$Self->{NotificationSubjectFollowUp} = 'You got follow up! (<OTRS_CUSTOMER_SUBJECT[10]>)';
$Self->{NotificationBodyFollowUp} = "
Hi <OTRS_USER_FIRSTNAME>,

you got a follow up!

<snip>
<OTRS_CUSTOMER_EMAIL[6]>
<snip>

http://$Self->{FQDN}/otrs/index.pl?Action=AgentZoom&TicketID=<OTRS_TICKET_ID>

Your OTRS Notification Master

";
[...]
```

12.11. Session management

The OTRS session management.

Example 12-13. Kernel/Config.pm - Session management

```
[...]
# -----#
# session settings                                     #
# -----#

# SessionModule (replace old SessionDriver!!!)
# (How should be the session-data stored?
# Advantage of DB is that you can split the
# Frontendserver from the DB-Server. fs is faster.)
$Self->{SessionModule} = 'Kernel::System::AuthSession::DB';
# $Self->{SessionModule} = 'Kernel::System::AuthSession::FS';
# $Self->{SessionModule} = 'Kernel::System::AuthSession::IPC';

# SessionCheckRemoteIP
# (If the application is used via a proxy-farm then the
# remote ip address is mostly different. In this case,
# turn of the CheckRemoteID. ) [1|0]
$Self->{SessionCheckRemoteIP} = 1;

# SessionDeleteIfNotRemoteID
# (Delete session if the session id is used with an
# invalied remote IP?) [0|1]
$Self->{SessionDeleteIfNotRemoteID} = 1;

# SessionMaxTime
# (Max valid time of one session id in second (8h = 28800).)
$Self->{SessionMaxTime} = 28800;

# SessionDeleteIfTimeToOld
# (Delete session's witch are requested and to old?) [0|1]
$Self->{SessionDeleteIfTimeToOld} = 1;

# SessionUseCookie
# (Should the session management use html cookies?
# It's more comfortable to send links ==> if you have a valid
# session, you don't have to login again.) [0|1]
# Note: If the client browser disabled html cookies, the system
# will work as usual, append SessionID to links!
$Self->{SessionUseCookie} = 1;
[...]
```

12.12. URL login and logout settings

Example 12-14. Kernel/Config.pm - URL login and logout settings

```
[...]
# -----#
# URL login and logout settings                       #
# -----#

# LoginURL
# (If this is anything other than "", then it is assumed to be the
# URL of an alternate login screen which will be used in place of
# the default one.)
$Self->{LoginURL} = "";
# $Self->{LoginURL} = 'http://host.example.com/cgi-bin/login.pl';

# LogoutURL
# (If this is anything other than "", it is assumed to be the URL
# of an alternate logout page which users will be sent to when they
# logout.)
$Self->{LogoutURL} = "";
# $Self->{LogoutURL} = 'http://host.example.com/cgi-bin/login.pl';
[...]
```

12.13. agent area default settings

Example 12-15. Kernel/Config.pm - agent area default settings

```
[...]
# -----#
# agent area default settings                                     #
# -----#

# ViewableTickets
# (The default viewable tickets a page.)
$Self->{ViewableTickets} = 25;

# ViewableTicketLines
# (Max viewable ticket lines in the QueueView.)
$Self->{ViewableTicketLines} = 18;

# ViewableTicketLinesZoom
# (Max viewable ticket lines in the QueueZoom.)
$Self->{ViewableTicketLinesZoom} = 6000;

# MaxLimit
# (Max viewable tickets a page.)
$Self->{MaxLimit} = 150;

# RefreshOptions
# (Refresh option list for preferences)
$Self->{RefreshOptions} = {
    " => 'off',
    2  => ' 2 minutes',
    5  => ' 5 minutes',
    7  => ' 7 minutes',
    10 => '10 minutes',
    15 => '15 minutes',
};

# Highlight*
# (Set the age and the color for highlighting of old queue
# in the QueueView.)
# highlight age1 in min
$Self->{HighlightAge1} = 1440;
$Self->{HighlightColor1} = 'orange';
# highlight age2 in min
$Self->{HighlightAge2} = 2880;
$Self->{HighlightColor2} = 'red';

# -----#
# AgentUtil                                                         #
# -----#

# default limit for Tn search
$Self->{SearchLimitTn} = 20;

# default limit for Txt search
$Self->{SearchLimitTxt} = 20;

# viewable ticket lines by search util
$Self->{ViewableTicketLinesBySearch} = 15;

# -----#
# Ticket stuff                                                         #
# (Viewable tickets in queue view)                                     #
# -----#
# ViewableLocks
# default: ["'unlock'", "'tmp_lock'"]
$Self->{ViewableLocks} = ["'unlock'", "'tmp_lock'"];

# ViewableStateType
# (see http://yourhost/otrs/index.pl?Action=AdminState -> StateType)
$Self->{ViewableStateType} = ['new', 'open', 'pending'];
```

```

# ViewableSenderTypes
# default:  ["'customer'"]
$Self->{ViewableSenderTypes} = ["'customer'"];
[...]
```

12.14. SpellChecker

The OTRS is able to use a spell checker like `ispell`(<http://fmg-www.cs.ucla.edu/fmg-members/geoff/ispell.html>) or `aspell`(<http://aspell.sourceforge.net/>).

Example 12-16. Kernel/Config.pm - SpellChecker

```

[...]
```

```

# SpellChecker
# (If ispell or aspell is available, then we will provide a spelling
# checker)
#
$Self->{SpellChecker} = "";
$Self->{SpellChecker} = '/usr/bin/ispell';
$Self->{SpellCheckerDictDefault} = 'english';
[...]
```

Possible agent dictionaries.

Example 12-17. Kernel/Config.pm - SpellChecker - Agent Preferences

```

[...]
```

```

$Self->{PreferencesGroups}->{SpellDict} = {
    Colum => 'Other Options',
    Label => 'Spelling Dictionary',
    Desc => 'Select your default spelling dictionary.',
    Type => 'Generic',
    Data => {
        # installed dict catalog (check your insalled catalogues,
        # e. g. deutsch ==> german!)
        # dict => frontend
        'english' => 'English',
        'deutsch' => 'Deutsch',
    },
    PrefKey => 'UserSpellDict',
    Activ => 1,
};
[...]
```

Maybe you want that the answer to the customer must be spell checked.

Example 12-18. Kernel/Config.pm - SpellChecker - Must be spell checked

```

# FrontendNeedSpellCheck
# (compose message must be spell checked)
$Self->{FrontendNeedSpellCheck} = 0;
```

Chapter 13. Cronjobs

OTRS needs a few system cronjobs to unlock tickets with lock timeouts, send reminder notifications and some other stuff.

13.1. How it works - setup

There are several OTRS default cronjobs in `$OTRS_HOME/var/cron/*.dist`. Make copies of all of the default cronjobs:

```
cd var/cron
for foo in *.dist; do cp $foo `basename $foo .dist`; done
```

Or if you are installing OTRS on a Windows system:

```
cd var/cron
copy *.dist *
```

Use `$OTRS_HOME/bin/Cron.sh {start|stop|restart}` to start or stop this cronjobs from `$OTRS_HOME/var/cron/*` (.dist will be ignored).

Note: Install this cronjobs as OTRS user.

13.2. Default cronjobs

13.2.1. aaa_base

Defined the email address of the OTRS admin which receive all cron error reports.

```
MAILTO="root@localhost"
```

13.2.2. pending_jobs

Processed every hour all reached "pending auto" tickets.

```
45 * * * * $HOME/bin/PendingJobs.pl >> /dev/null
```

13.2.3. postmaster

Cleanup at 00:10 of all unprocessed emails which are received via procmail.

```
10 0 * * * $HOME/bin/otrs.cleanup >> /dev/null
```

13.2.4. postmaster_pop3

Fetch emails via PostMasterPOP3.pl every 10 minutes.

```
*/10 * * * * $HOME/bin/PostMasterPOP3.pl >> /dev/null
```

13.2.5. unlock

Unlock every hour old locked tickets.

```
35 * * * * $HOME/bin/UnlockTickets.pl --timeout >> /dev/null
```

13.2.6. stats

Create every day new graphics (default: not active).

```
50 23 * * * * $HOME/bin/mkStats.pl >> /dev/null
```

13.2.7. session

Delete every six hours old session ids from database, filesystem or RAM.

```
55 */6 * * * * $HOME/bin/DeleteSessionIDs.pl --expired >> /dev/null
```

13.2.8. rebuild_ticket_index

Rebuild of ticket index every day (needed for Kernel::System::Ticket::IndexAccelerator::StaticDB - see also chapter "Performance Tuning").

```
01 01 * * * * $HOME/bin/RebuildTicketIndex.pl >> /dev/null
```

13.2.9. generic_agent

Execute GenericAgent.pl every 20 minutes (see also chapter "Generic-Agent").

```
*/20 * * * * $HOME/bin/GenericAgent.pl >> /dev/null
```

Chapter 14. Generic-Agent

The Generic-Agent (bin/GenericAgent.pl) is a command line program which can do some actions (like move, add note, set state, set owner) on tickets.

14.1. Config File

Kernel/Config/GenericAgent.pm is the default config file which must be copied (cp Kernel/Config/GenericAgent.pm.dist Kernel/Config/GenericAgent.pm).

14.2. Examples

Now we want to go through a few examples. There is also an example config file (Kernel/Config/GenericAgent.pm.examples).

Here are all possible options:

```
'name of job' => {
    # get all tickets with these properties
    Queue => 'system queue',
    States => ['new', 'open'],
    Locks => ['unlock'],
    # or escalation tickets
    Escalation => 1,

    # new ticket properties (no option is required, use just the options
    # which should be changed!)
    New => {
        # new queue
        Queue => 'new system queue',
        # possible states (closed successful|closed unsuccessful|open|new|removed)
        State => 'closed successful',
        # new ticket owner (if needed)
        Owner => 'root@localhost',
        # if you want to add a Note
        Note => {
            From => 'GenericAgent',
            Subject => 'Your subject!',
            Body => 'Some comment!',
            ArticleType => 'note-internal', # note-internal|note-external|note-report
        },
        # your program (/path/to/your/program) will be executed like
        # "/path/to/your/program $TicketNumber $TicketID" ARG[0] will
        # be the ticket number and ARG[1] the ticket id
        CMD => '/path/to/your/program',
        # DELETE ticket from database and filesystem
        Delete => 1,
    },
},
```

14.2.1. close spam tickets

If you get spam tickets in your system and you want to spend much time on it, create a queue (named spam) and just move the spam tickets into this queue. If the bin/GenericAgent.pl runs, all open tickets in the queue spam will be closed by the bin/GenericAgent.pl.

Example for Kernel/Config/GenericAgent.pm

```
[...]
# --
# [name of job] -> close all tickets in queue spam
# --
'close spam' => {
    # get all tickets with this properties
    Queue => 'spam',
    States => ['new', 'open'],
```

```

Locks => ['unlock'],
# new ticket properties (no option is required, use just the options
# which should be changed!)
New => {
  # new queue
  Queue => 'spam',
  # possible states (closed successful|closed unsuccessful|open|new|removed)
  State => 'closed successful',
  # new ticket owner (if needed)
  Owner => 'root@localhost',
  # if you want to add a Note
  Note => {
    From => 'GenericAgent',
    Subject => 'spam!',
    Body => 'Closed by GenericAgent.pl because it is spam!',
  },
},
},
[...]
```

14.2.2. delete tickets

If you want to delete (means delete ticket from database and filesystem) a ticket from a queue use this.

Exampe for Kernel/Config/GenericAgent.pm

```

[...]
```

```

# --
# [name of job] -> close and delete all tickets in queue delete
# --
'delete' => {
  # get all tickets with this properties
  Queue => 'delete',
  States => ['new', 'open'],
  Locks => ['unlock'],
  # new ticket properties (no option is required, use just the options
  # witch should be changed!)
  New => {
    # DELETE!
    Delete => 1,
  },
},
[...]
```

14.2.3. move tickets from tricky to experts

Moved ticket from tricky to experts queue and add note.

Exampe for Kernel/Config/GenericAgent.pm

```

[...]
```

```

# --
# [name of job] -> move all tickets from tricky to experts
# --
'move tickets from tricky to experts' => {
  # get all tickets with this properties
  Queue => 'tricky',
  States => ['new', 'open'],
  Locks => ['unlock'],
  # new ticket properties
  New => {
    Queue => 'experts',
    Note => {
      From => 'GenericAgent',
      Subject => 'Moved!',
      Body => 'Moved from "tricky" to "experts" because it was not possible to find a sollution!',
    },
  },
},
[...]
```

```

        ArticleType => 'note-internal', # note-internal|note-external|note-report
    },
},
},
[...]
```

14.2.4. move escalation ticket to experts and execute CMD

If there is a escalation ticket, move it to experts and execute a command.

Exampe for Kernel/Config/GenericAgent.pm

```

[...]
```

```

# --
# [name of job] -> move all tickets from xyz to experts
# --
'move escalation ticket to experts and execute CMD' => {
    # get all tickets with this properties
    Queue => 'xyz',
    Escalation => 1,
    # new ticket properties
    New => {
        Queue => 'experts',
        # your program (/path/to/your/program) will be executed like
        # "/path/to/your/program $TicketNumber $TicketID" ARG[0] will
        # be the ticket number and ARG[1] the ticket id
        CMD => '/path/to/your/program',
    },
},
[...]
```

Chapter 15. Language translations

The OTRS webfrontend supports different frontend languages. The language translation files are located under Kernel/Language/*.pm.

15.1. New translation files

If you want to translate OTRS in a new language, you have to do five steps:

- Take the current German translation (Kernel/Language/de.pm) from CVS (<http://cvs.otrs.org/>).
- Change the package name (z. g. "package Kernel::Language::de;" to "package Kernel::Language::fr;") and translate each word/sentence.
- Add the new language translation to the system via adding "\$Self->{DefaultUsedLanguages}->{fr} = 'French';" to your Kernel/Config.pm.
- If you use mod_perl, restart your webserver and you will have the new language.
- Send the new translation file to feedback at otrs.org - Thanks!

Now you can select the new language in your preferences screen.

Example 15-1. Kernel/Language/de.pm - Old file

```
# --
# Kernel/Language/de.pm - provides de language translation
# Copyright (C) 2002-2003 Martin Edenhofer xxxxxxxxxxxxxxxxxxxxxx
# --
# $Id: language-translation.sgml,v 1.9 2003/01/23 20:00:06 martin Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY. For details, see
# the enclosed file COPYING for license information (GPL). If you
# did not receive this file, see http://www.gnu.org/licenses/gpl.txt.
# --
package Kernel::Language::de;

use strict;

use vars qw($VERSION);
$VERSION = '$Revision: 1.9 $';
$VERSION =~ s/^\.*:\s(\d+\.\d+)\s.*\$/\$1/;

# --
sub Data {
    my $Self = shift;
    my %Param = @_;
    my %Hash = ();

    # $$START$$
    # Last translation Fri Jan 3 20:39:15 2003 by

    # possible charsets
    $Self->{Charset} = ['iso-8859-1', 'iso-8859-15', ];
    # date formats (%A=WeekDay;%B=LongMonth;%T=Time;%D=Day;%M=Month;%Y=Year;)
    $Self->{DateFormat} = '%D.%M.%Y %T';
    $Self->{DateFormatLong} = '%A %D %B %T %Y';
    $Self->{DateInputFormat} = '%D.%M.%Y - %T';

    %Hash = (
        # Template: AAABase
        ' 2 minutes' => ' 2 Minuten',
        ' 5 minutes' => ' 5 Minuten',
        ' 7 minutes' => ' 7 Minuten',
        '10 minutes' => '10 Minuten',
        '15 minutes' => '15 Minuten',
        'AddLink' => 'Link hinzufügen',
        'AdminArea' => 'AdminBereich',
        'all' => 'alle',
```

```

    'All' => 'Alle',
    'Attention' => 'Achtung',
[...]
```

To:

Example 15-2. Kernel/Language/fr.pm - New file

```

# --
# Kernel/Language/fr.pm - provides fr language translation
# Copyright (C) 2002 Bernard Choppy xxxxxxxxxxxxxxxxxxxxxxxx
# Copyright (C) 2002-2003 Nicolas Goralski xxxxxxxxxxxxxxxxxxxxxxxx
# --
# $Id: language-translation.sgml,v 1.9 2003/01/23 20:00:06 martin Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY. For details, see
# the enclosed file COPYING for license information (GPL). If you
# did not receive this file, see http://www.gnu.org/licenses/gpl.txt.
# --
package Kernel::Language::fr;

use strict;

use vars qw($VERSION);
$VERSION = '$Revision: 1.9 $';
$VERSION =~ s/^\.*:\s(\d+\.\d+)\s.*\$/\$/;
# --
sub Data {
    my $Self = shift;
    my %Param = @_;
    my %Hash = ();

    # $$START$$
    # Last translation Fri Jan 3 20:40:04 2003 by

    # possible charsets
    $Self->{Charset} = ['iso-8859-1', 'iso-8859-15', ];
    # date formats (%A=WeekDay;%B=LongMonth;%T=Time;%D=Day;%M=Month;%Y=Year;)
    $Self->{DateFormat} = '%D.%M.%Y %T';
    $Self->{DateFormatLong} = '%A %D %B %T %Y';
    $Self->{DateInputFormat} = '%D.%M.%Y - %T';

    %Hash = (
    # Template: AAABase
    ' 2 minutes' => ' 2 minutes',
    ' 5 minutes' => ' 5 minutes',
    ' 7 minutes' => ' 7 minutes',
    '10 minutes' => '10 minutes',
    '15 minutes' => '15 minutes',
    'AddLink' => 'Ajouter un lien',
    'AdminArea' => 'Zone d\'administration',
    'all' => 'tout',
    'All' => 'Tout',
    'Attention' => 'Attention',
[...]
```

Chapter 16. Customize the frontend

It's possible to customize the frontend independently from OTRS releases. How? It's quite simply. The magic key is dtl (Dynamic Template Language). All frontend masks are located under `~otrs/Kernel/Output/HTML/<THEME>/*.dtl`. Default is the "Standard" Theme.

So you have the power to customize each OTRS side like you want! Or to create new themes.

16.1. The dtl syntax

Comment

Comment is a simple '#'.

```
# --
# this is a comment
# --
```

Set a variable

```
<dtl set $Data{"Test1"} = "German">
```

Note: `$Data{"xyz"}` exists only the current dtl file and `$Env{"xyz"}` exists the whole dtl files. New: `$Config{"xyz"}` is not read only anymore and exists the whole program! (2002-05-22 / 0.5 BETA5)

Print a variable

To print a variable on the screen, use simply:

```
$Data{"xyz"} or $Env{"xyz"}
```

Text translations

```
$Text{"This should be translated"}
```

Take care, that the translation exists in the `"$HOME_OTRS/Kernel/Language/*.pm"` files. If there isn't a translation, the given text will be shown.

Condition

```
<dtl if ($Text{"Lock"} eq "Lock") { $Data{"FrontendLanguage"} = "English"; }>
```

It's only possible to store things into `$Data{"xyz"}` and `$Env{"xyz"}`.

Get a config option - \$Config{}

```
$Config{"Sendmail"}
```

Common environment variables - \$Env{}

```
$Env{"SessionID"} --> the current session id
$Env{"Time"} --> the current time e. g. 'Thu Dec 27 16:00:55 2001'
$Env{"CGIHandle"} --> the current CGI handle e. g. 'index.pl'
$Env{"UserCharset"} --> the current site charset e. g. 'iso-8859-1'
$Env{"Baselink"} --> the baselink --> index.pl?SessionID=...
$Env{"UserFirstname"} --> e. g. Dirk
$Env{"UserLastname"} --> e. g. Hohndel
$Env{"UserLogin"} --> e. g. mgg@x11.org
$Env{"UserIsGroup[users]"} = Yes --> user groups (useful for own links)
$Env{"UserIsGroup[admin]"} = Yes
$Env{"Action"} --> the current action
```

```
$Env{"Subaction"} --> the current subaction
```

System calls

To get the output of a system command use:

```
# execute system call
<dtl system-call $Data{"uptime"} = "uptime">

# print
$Data{"uptime"}

or

# execute system call
<dtl system-call $Data{"procinfo"} = "procinfo | head -nl ">

# print
$Data{"procinfo"}
```

Examples

```
# set variable
<dtl set $Data{"Test1"} = "English">

# print variable
Echo: $Data{"Test1"}

# condition
<dtl if ($Text{"Lock"} ne "Lock") { $Data{"Test2"} = "Not English!"; }>

# print result
Result: $Data{"Test1"}

or

# translation test
Lock: $Text{"Lock"}

# config options
Sendmail: $Config{"Sendmail"}
```

16.2. Examples of dtl files

16.2.1. Motd.dtl

If you want to customize this file use the `~otrs/Kernel/Output/HTML/Standard/Motd.dtl` file.

```
<p>
This is the message of the day. You can edit this in Kernel/Output/HTML/Standard/Motd.dtl.
</p>
```

16.2.2. Login.dtl

This is an example of the login screen. If you want to customize this screen use the `~otrs/Kernel/Output/HTML/<THEME>/Login.dtl` file.

```
# --
# http headers
# --
```

```

Content-Type: text/html; charset=${Env{"UserCharset"}};
X-Powered-By: OTRS - Open Ticket Request System (http://otrs.org)

# --
# html comment
# --

<!-- OTRS: Copyright 2002, OTRS Project. This Software is under the GPL. -->
<!--      Web: http://otrs.org/ - Lists: http://lists.otrs.org/      -->
<!--      GNU Public License: http://www.gnu.org/licenses/gpl.txt  -->

# --
# set some html variables
# --
<dtl set ${Env{"BGCOLOR"}} = "#BDDDFD">
<dtl set ${Env{"BGTableColor0"}} = "#000000">
<dtl set ${Env{"BGTableColor1"}} = "#FFFFFF">
<dtl set ${Env{"BGTableColor2"}} = "#EEEEEE">
<dtl set ${Env{"FontColor0"}} = "#000000">
<dtl set ${Env{"FontColor1"}} = "#FFFFFF">
<html>
<head>
  <title>OTRS :: ${Text{"$Data{"Title"}}}</title>
</head>
<!-- end header -->
<body bgcolor=${Env{"BGCOLOR"}} text=${Env{"FontColor0"}}>

<center>

<p>
<font color="red">${Data{"Message"}}</font>
</p>

<p>
<form action=${Env{"CGIHandle"}} method="post" enctype="application/x-www-form-urlencoded">
<input type="hidden" name="Action" value="Login">

<table border="0" cellspacing="0" cellpadding="3" width="240">
<tr bgcolor=${Env{"BGTableColor0"}}>
  <td align="center"><font color=${Env{"FontColor1"}}><b>${Text{"Welcome to OTRS"}}</b></font></td>
</tr>
<tr>
  <td align="center" bgcolor=${Env{"BGTableColor1"}}>
    <table cellspacing="8" cellpadding="2">
      <tr>
        <td>Username:</td>
        <td><input type="text" name="User" value=${Data{"User"}} size="18"></td>
      </tr>
      <tr>
        <td>Password:</td>
        <td><input type="password" name="Password" size="18"></td>
      </tr>
    </table>
    <input type="submit" value=${Text{"Login"}}>
  </td>
</tr>
</table>
</form>
</p>

# --
# Message of the day data!
# --
${Data{"Motd"}}

</center>

</body>
</html>

```

16.2.3. Header.dtl

This is the default HTML header of each OTRS sides. If you want to customize this screen use the `~otrs/Kernel/Output/HTML/<THEME>/Login.dtl`

```
# --
# http headers
# --
Content-Type: text/html; charset=$Env{"UserCharset"};

# --
# html comment
# --

<!-- OTRS: Copyright 2002, OTRS Project. This Software is under the GPL. -->
<!--      Web: http://otrs.org/ - Lists: http://lists.otrs.org/      -->
<!--      GNU Public License: http://www.gnu.org/licenses/gpl.txt  -->
# --
# set some html variables
# --
<dtl set $Env{"BGCOLOR"} = "#BDDDF" >
<dtl set $Env{"BGTableColor0"} = "#000000" >
<dtl set $Env{"BGTableColor1"} = "#FFFFFF" >
<dtl set $Env{"BGTableColor2"} = "#EEEEEE" >
<dtl set $Env{"FontColor0"} = "#000000" >
<dtl set $Env{"FontColor1"} = "#FFFFFF" >
<dtl set $Env{"Box0"} = "[ ">
<dtl set $Env{"Box1"} = "]">
# --
# check refresh
# --
<dtl if ($Data{"Refresh"} ne "") { $Data{"MetaHttpEquiv"} = "<meta http-equiv='refresh' content='$Data{"Refresh"}'"/>
# --
<html>
<head>
    <title>OTRS :: $Text{"$Data{"Title"}}</title>
</head>
<!-- end header -->
```

16.3. Create a new theme

To create a new theme, create a new theme directory (`mkdir ~otrs/Kernel/Output/HTML/NewTheme/`). Copy an existing theme into the new directors (`cp ~otrs/Kernel/Output/HTML/Standart/*.dtl ~otrs/Kernel/Output/HTML/NewTheme/`).

Modify the dtl files like you want.

Add a new database entry for this theme. You have to do this via SQL.

```
shell:~> mysql -u root -p some-pass otrs
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 3.23.48-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> INSERT INTO theme
->      (theme, valid_id, create_by, change_by, change_time)
->      VALUES
->      ('NewTheme', 1, 1, 1, current_timestamp);
mysql>
```

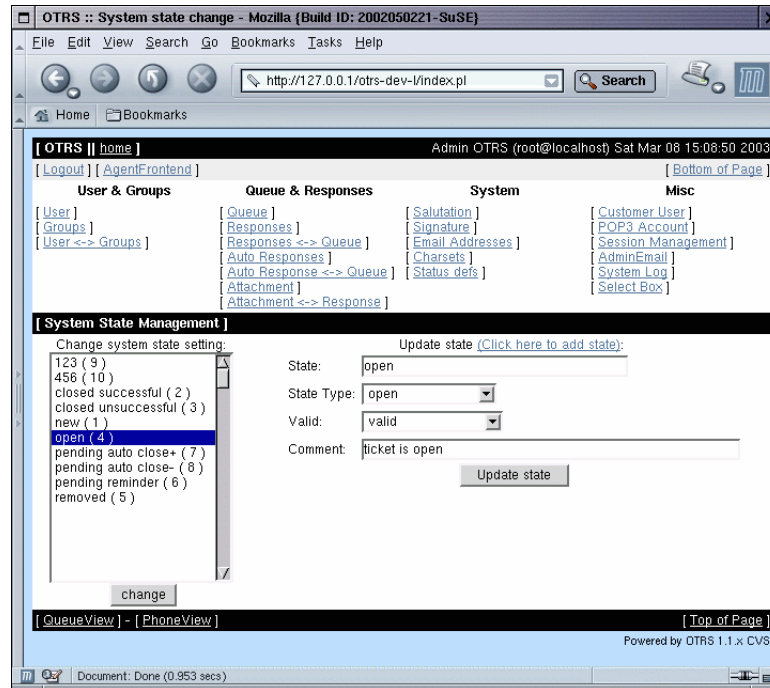
Ready. Goto the preferences page and change your theme.

Chapter 17. Customize Ticket State (min. OTRS 1.1)

It's possible to change or add ticket states. There are two important options, state-name and state-type.

- The default state-types are 'new', 'open', 'closed', 'pending reminder', 'pending auto' and 'removed'.
- Each state needs to have a name (state-name) and a type (state-type). The name is free settable.

Use the Admin-Interface (Status defs) if you want to change or add ticket states.



Take care that you also update some config options if you changed/rename the state "new". Add to you Kernel/Config.pm:

```
[...]  
# PostmasterDefaultState  
# (The default state of new tickets.) [default: new]  
$Self->{PostmasterDefaultState} = 'new';  
[...]
```

Take care that you also update some config options if you changed/rename the state "open". Add to you Kernel/Config.pm:

```
[...]  
# default phone new state  
$Self->{PhoneDefaultNewNextState} = 'open';  
  
# PostmasterFollowUpState  
# (The state if a ticket got a follow up.) [default: open]  
$Self->{PostmasterFollowUpState} = 'open';  
[...]
```

Chapter 18. Customize Ticket Priority (min. OTRS 1.1)

If you want to customize the system ticket priorities follow the next steps (there is no existing webfrontend).

- Get the current priorities:

```
mysql> select id, name from ticket_priority;
```

```
+-----+-----+
| id | name          |
+-----+-----+
| 1 | 1 very low   |
| 2 | 2 low        |
| 3 | 3 normal     |
| 4 | 4 high       |
| 5 | 5 very high  |
+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql>
```

Important: The "id" gives the order of the priority. -=> 1 = min, 5 (or higher) = highest

The number in front of the state name is to show the order correctly in the webinterface.

- Update the priorities via SQL. e. g.

```
mysql> UPDATE ticket_priority SET name = '3 default' WHERE id = 3;
```

And the "normal" priority will be in future named "3 default".

- Take care that you also update the default priority (if you changed the name) in your Kernel/Config.pm

```
[...]
# PostmasterDefaultPriority
# (The default priority of new tickets.) [default: '3 normal']
$self->{PostmasterDefaultPriority} = '3 default';
[...]
# default phone priority
$self->{PhoneDefaultPriority} = '3 default';
[...]
# CustomerDefaultPriority
# (default priority of new customer tickets)
$self->{CustomerDefaultPriority} = '3 default';
[...]
```

Chapter 19. Multi Hosting

OTRS is able to manage different groups and domains.

19.1. Multi Group

This is an example setup for one company with a helpdesk, sales and billing department. Each department should not see tickets of the other department. This means nobody from helpdesk should have access to the billing tickets. Nobody the one department can access to a queue of the others.

Add the following to Kernel/Config.pm

```
# MoveIntoAllQueues -> useful for ASP
# (Possible to move in all queue? Not only queue which
# the own groups) [1|0]
$Self->{MoveIntoAllQueues} = 0;

# ChangeOwnerToEveryone -> useful for ASP
# (Possible to change owner of ticket ot everyone) [0|1]
$Self->{ChangeOwnerToEveryone} = 0;

# ShowCustomerSelection
# (show customer selection in phone and change customer view
# - disable this for ASP!) [0|1]
$Self->{ShowCustomerSelection} = 0;

# PhoneViewASP -> useful for ASP
# (Possible to create in all queue? Not only queue which
# the own groups) [0|1]
$Self->{PhoneViewASP} = 1;
```

System email addresses (1/2): You need to add new system email addresses for each department (e. g. helpdesk@example.com, sales@example.com and billing@example.com). Note: You can't set the right queue because you need to create the queue at first - this will be done later.

Groups: Add a new group for each department. In our case add a group helpdesk, sales and billing.

Queue: Add queue for each department (take care of the naming - e. g. group-topic). In our case add queue helpdesk-raw, helpdesk-products, sales-raw, sales-products, billing-raw and billing-claim and set the group of each queue to the right.

System email addresses (2/2): Update your new system email addresses (e. g. helpdesk@example.com, sales@example.com and billing@example.com) with the right queue. Note: Of course you need to make sure that the added system email addresses are routed into your system (e. g. via admin interface ==> POP3 accounts - see chapter "Receiving emails")

User: Add the users and put each user into the wanted group. Means sales users to the sales group, helpdesk users to the helpdesk group and billing users to the billing group.

Note: If you add one user of the departments to the admin group, then the user can change all settings of the system. Means the user can also add/modify user/response/... of the other groups. ==> Normally this admin changes should be done by an admin.

Your multi group setup is ready.

19.2. Multi Setup

If you want a productiv and a test system or you want to host more the one OTRS systems on one machine you have to make sure that each OTRS has a unique SystemID.

Add the following to each Kernel/Config.pm

```
# SystemID
# (The identify of the system. Each ticket number and
# each http session id starts with this number)
$self->{SystemID} = n;
```

where n is a reasonable low integer that is unique across the system! Also note that you cannot run more than one OTRS System through mod_perl-1 (one httpd). In case you want more the one OTRS System on one machine with mod_perl-1, start a new httpd on an other ip-address or port. mod_perl-2 can handle this with one httpd.

Note: If you use the "scripts/apache-perl-startup.pl" for mod_perl, change the "use lib" directory (e. g. /opt/otrs or /opt/otrs2)!

Chapter 20. Performance Tuning

An exhaustive list of various techniques you might want to use to get the most performance possible out of your OTRS system: configuration, coding, memory use and more.

20.1. OTRS

There are several options to improve the performance via OTRS.

20.1.1. TicketIndexModule

You have two backend modules for the ticket index.

Kernel/Config.pm

```
[ ... ]
    $Self->{TicketIndexModule} = 'Kernel::System::Ticket::IndexAccelerator::RuntimeDB';
[ ... ]
```

- Kernel::System::Ticket::IndexAccelerator::RuntimeDB (default), generate each queue view on the fly from ticket table. You will not have performance trouble till ~ 50.000 tickets in your system.
- Kernel::System::Ticket::IndexAccelerator::StaticDB, the most powerfull module, it should be used over 60.000 tickets in a system - use a extra ticket_index table, works like a view. Use bin/RebuildTicketIndex.pl for initial index build.

20.1.2. TicketStorageModule

You have two backend modules for the ticket/article storage .

Kernel/Config.pm

```
[ ... ]
    $Self->{TicketStorageModule} = 'Kernel::System::Ticket::ArticleStorageDB';
[ ... ]
```

- Kernel::System::Ticket::ArticleStorageDB (default), store attachments and co into the database. Note: Don't use it for larger setups.
Pro: If your webserver user isn't the otrs user, use this module to have no file permission problems.
Contra: It's not really nice to store attachments in your database. Take care that your database is able to store large objects. E. g. MySQL (config) "set-variable = max_allowed_packet=8M" to store 8 MB objects (default is 2M).
- Kernel::System::Ticket::ArticleStorageFS, store attachments and co in local file system. Note: Use it for larger setups.
Pro: Faster!
Contra: Your webserver user should be the otrs user (file system permissions!).

20.2. Database

This depends on the database you are using. If you have trouble, read the documentation for your database or ask your database admin.

20.3. Webserver

Of course you should use mod_perl (<http://perl.apache.org/>). It's much faster (~ * 100) than pure cgi. But needs more RAM. So your httpd with mod_perl will be about 16 MB (~10MB shared).

And you can have establish datababase connections on process startup (httpd). This saves also time (see README.webserver).

Use the scripts/apache-perl-startup.pl startup script for your mod_perl webserver to be faster (see README.webserver).

If you have a really large installation (over 1000 new tickets a day and over 40 Agents) is a good idea to read "Choosing the Right Strategy" (<http://perl.apache.org/docs/1.0/guide/strategy.html>).

Anyway, if your bandwidth is a little bit small use mod_gzip (http://www.schroepl.net/projekte/mod_gzip/). If you have a html page with 45k, mod_gzip compress it. The page will be about 7k. That's really nice.

Chapter 21. Troubleshooting

We split this section into different parts per distribution. Of course you can set up your own system with your own linux system. But please understand that we can not provide any support for that. OTRS is much too complicated and has to many links to other components of the system that we are lucky to have access to mechanisms like RPM. We will try to support as many platforms as possilbe but time is a valuable resource. ;-)

21.1. General problems with OTRS on SuSE Linux

The very first step should be a visit to <http://www.suse.de/de/support/download/updates/> which is the page where you can find the latest patches and updates for your SuSE Linux. Please check especialy for fixes about Apache, MySQL, Perl and of course OTRS. YaST2 users should be able to use the Yast Online Update mechanism. Otherwise download the rpm files and deinstall the old package with "rpm --nodeps -e otrs" and install them th new with "rpm -i foo.rpm"

The second step should be a visit to the *OTRS Homepage*(<http://www.otrs.org/>). We will provide the very latest fixes and howtos there.

The third step is writting an e-mail to the developer team of OTRS. We will be more than happy to provide you with any support you need.

21.1.1. SuSE Linux 8.0

Unfortunately the otrs.rpm which is on the SuSE 8.0 distribution is a buggy one (it was not a SuSE mistake). Please download the newest version from our Homepage(<http://www.otrs.org/>) and install it with YaST or manually by

```
# deinstall old package
shell> rpm --nodeps -e otrs
# install new package
shell> rpm -i new-otrs.rpm
```

21.1.2. SuSE Linux and Postfix

Postfix isn't configured with Procmail out of the box. You have to enable Procmail in /etc/postfix/main.cf:

```
[...]
mailbox_command = /usr/bin/procmail
[...]
```

And to restart the Postfix daemon (rcpostfix restart).

21.2. General problems with OTRS on other distributions (e.g. Redhat)

Frankly we have not tried to install it on a Redhat yet. It should be a problem but there is no ready to use RPM. We are working on it. Same for other distributions.

21.3. Problems with Apache

Most people who have problems with the Apache did build their own very special Apache. Of course you are welcome to do so but in case you run into trouble we suggest to your the vanilla version which is provided by your favorite distribution.

21.3.1. Internal Server Error

In this case check the syntax of the index.pl file to find the error:

```
shell:~ # cd ~otrs
```

It's important to be in the \$HOME of the otrs user.

```
shell:/opt/otrs # perl -cw bin/cgi-bin/index.pl
bin/cgi-bin/index.pl syntax OK
shell:/opt/otrs #
```

If you get an error message, in most cases you have to install missing perl modules from CPAN(<http://www.cpan.org/>).

Install CPAN modules via cpan shell:

```
shell:~ # perl -MCPAN -e shell;
-- (may you have to configure cpan first) --

cpan shell -- CPAN exploration and modules installation (v1.59_54)
ReadLine support enabled

cpan> install Digest::MD5
[installing Digest::MD5 module]
cpan>
```

Check the perl syntax again (perl -cw bin/cgi-bin/index.pl) and install further missing modules if necessary.

21.3.2. Error: Can't connect to database!

If your browser get the message "Error: Can't connect to database!" after you changed the database settings, restart the webserver. mod_perl reads the ~otrs/Kernel/Config.pm only on startup.

If the "Error: Can't connect to database!" message is still there, check the error log of your webserver (e. g. /var/log/httpd/error_log).

21.3.3. FreeBSD, PostgreSQL and Apache (install_driver(Pg) failed)

Everything is working fine just if I want to use the web interface I get always: [...] Software error: install_driver(Pg) failed: [Thu Sep 19 16:52:18 2002] index.pl: [Thu Sep 19 16:52:18 2002] index.pl: Can't load '/usr/local/lib/perl5/site_perl/5.005/i386-freebsd/auto/DBD/Pg/Pg.so' for module DBD::Pg: Shared object "libpq.so.2" not found at /usr/libdata/perl/5.00503/DynaLoader.pm line 169. [Thu Sep 19 16:52:18 2002] index.pl: [Thu Sep 19 16:52:18 2002] index.pl: [Thu Sep 19 16:52:18 2002] index.pl: at (eval 124) line 3 Perhaps a required shared library or dll isn't installed where expected at /usr/local/otrs/bin/cgi-bin/../../Kernel/System/DB.pm line 67 [...]

To solve this problem, you can tell the Apache httpd.conf to include the environmental variable LD_LIBRARY_PATH=/usr/local/pgsql/lib. Put this line to httpd.conf and then restart "SetEnv LD_LIBRARY_PATH /usr/local/pgsql/lib" and then restart (source <http://lists.otrs.org/pipermail/otrs/2002-September/000248.html>).

21.4. Problems with MySQL

Please doublecheck all passwords. Mostly people setup the system with a wrong password. In this case it is the easiest way to re-setup the system.

21.4.1. Check the database connect

To check the database connect use '~otrs/bin/CheckDB.pl'. Is the output "It looks Ok!", you don't have problems with MySQL. If not, read the error message.

21.4.2. Access denied for user: 'otrs@localhost'

Check the password in Kernel/Config.pm for the database user. If you don't know the database password for the database otrs user, set it new:

```
shell> mysql -u root -p
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
```

Your MySQL connection id is 37 to server version: 3.23.48-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

```
mysql> GRANT ALL PRIVILEGES ON otrs.* TO otrs@localhost IDENTIFIED BY "yourpw" WITH GRANT OPTION;
mysql>
```

Reload the grant tables of your mysql-daemon:

```
$shell> mysqladmin -u root -p reload
```

(or restart your mysql-daemon) and change the new otrs password in Kernel/Config.pm.

Note: If you use mod_perl, restart the webserver, because mod_perl is reading the perlcode (incl. Kernel/Config.pm) on startup.

21.4.3. Lost root password of MySQL

Follow: http://www.mysql.com/documentation/mysql/bychapter/manual_Problems.html#Resetting_permissions

21.4.4. <http://localhost/otrs/installer.pl>

If you want to set up the otrs database again, you can call the setup program by browsing to <http://localhost/otrs/installer.pl>.

Note: Just for SuSE Linux RPMs.

21.5. Problems - misc

21.5.1. Problems with receiving emails

If everything is working, but you can't see the incoming emails, check if the `~otrs/bin/PostMaster.pl` program is working correctly. Pipe an example email through the `PostMaster.pl`. There are two example emails in the default OTRS package (`~otrs/doc/test-email-*.box` or `/usr/share/doc/packages/otrs/test-email-*.box`).

Example:

```
shell:~ # cat /usr/share/doc/packages/otrs/test-email-1.box | /opt/otrs/bin/PostMaster.pl
shell:~ #
```

If there is an error message, then you will have to fix it. If not and you see this email in the postmaster queue, the procmail program isn't configured with your MTA (check the MTA log file, e. g. `/var/log/mail`) or the `$HOME` of the otrs user isn't correct (check `$HOME` and `$HOME/.procmailrc`).

21.5.2. Lost root password of OTRS

I lost the root password of OTRS ("Login failed! Your username or password was entered incorrectly.").

Reset of the OTRS root password:

```
shell:~ # mysql -u root -p otrs
mysql> UPDATE system_user SET pw='roK20XGbWESM' where login='root@localhost';
mysql>
```

And the password of user 'root@localhost' will be 'root'. The password is crypted like 'man 3 crypt' with `crypt()`.

Chapter 22. FAQ

1. I installed the otrs.rpm from SuSE 8.0. But it doesn't work. Why?

The default SuSE 8.0 rpm is buggy (which is not a SuSE mistake). Please visit our website(<http://www.otrs.org/>) and download the newest rpm. You can install the rpm with YaST or manually

```
rpm -Uvh otrs.rpm
```

2. What is OTRS?

OTRS is a Ticket Request System with many features to manage customer telephone calls and e-mails.

3. What stands OTRS for?

Open Trouble-Ticket Request System.

4. What does OTRS cost?

Nothing, it's GPL(<http://www.gnu.org/copyleft/gpl.txt>).

5. This document refers to agents. What is an agent in the context of OTRS?

An Agent is a human being who works with the system. User would be an alternative term.

Geeks: Of course a script could act like an agent. Feel free to do so!

6. How can I delete a user, who is no longer needed? (asked by Andreas Haase - Wed, 21 Aug 2002 11:17:01 +0200 (CEST))

OTRS is working with database id references. If you would delete a user (or queue, ...) from the database, you delete the reference info. Important infos like owner or queue.

Set the data record to invalid and the record is still visible (e. g. for ticket history, ...) but not active for the application.

7. Can OTRS receive email?

Yes, (with MIME support).

The \$HOME/bin/PostMaster.pl program receives the emails and sorts the email to the right ticket or queue.

8. Can OTRS send email?

Yes. Autoresponders per queue or per X-Header and standard responders via mouse click.

Note: You can add/delete/modify the system email addresses at AdminView::System.

9. Fulltext Index Searching?

Yes.

10. Is OTRS multi user and multi group able?

Yes, of course.

11. Is it possible to use different domains?

Yes, you can use different emails 'support@yourdomain.com', 'sales@yourdomain.com' and different domains 'marketing@clientdomain.com' with one system!

12. Can I use OTRS only with SuSE Linux?

No, but we developed it on a SuSE Linux and frankly we did not do much testing on a Redhat, Debian, ...

But we will provide additional installation information on <http://www.otrs.org/>. And we are more than happy to receive some feedback from you about how you installed OTRS on other platforms.

13. What software will be needed?

Minimum: Perl5(<http://www.perl.com/>) (with a few CPAN(<http://www.cpan.org/>) modules, more: INSTALL), MySQL(<http://www.mysql.com>) and Apache(<http://www.apache.org>).

But again, your are on the safe side by using SuSE Linux. It will provide you with all the needed stuff.

14. How stable is OTRS?

Please be aware of the fact that you are dealing with a beta-version. New versions are announced on <http://www.otrs.org/>. But never the less it is quite a stable system and you shouldn't run in any trouble. But we can not guarantee it!

15. What hardware do I need?

We suggest an IBM s390. *SCNR* ;-)

Some of our test enviroments are Pentium II 300 with 64 MB RAM and they do a pretty good job. Of course the more RAM and the faster the CPU the better.

16. How does OTRS scale and how big can it become?

This is depending on the hardware and the enviroment you are using. At the moment OTRS is a one box system. With little work you can set up a webserver-cluster and you can split the database to a seperate box. We are planing to support some sort of clustering mechanism. But this is not the highest priority for the development.

There are OTRS installations which handle 20,000 tickets (e-mails) a day and don't show any sign of stress.

17. Can I use my nice Oracle or DB2?

At the moment we only support MySQL(<http://www.mysql.com>) as the default database. Frankly we do trust in a MySQL(<http://www.mysql.com>) as much as in a DB2 (for this application). Anybody who is willing to spend some time to port it to other databases is more than welcome to the OTRS team!

Note: The code is designed to support different databases!

18. Which programming language is used?

Perl(<http://www.perl.com/>) OO, SQL and dtl.

19. What is the default admin account?

User: root@localhost Password: root

Of course it is a very good idea to change this default password!

20. Works OTRS with mod_perl(<http://perl.apache.org/>)?

Of course.

21. Is it possible to customize OTRS?

Of course. You can customize OTRS like you want. Take advantage of the dtl (dynamic template language) to customize the OTRS frontend release independently (more: README.dtl)!

22. Can I install OTRS on a Windows box?

Theoraticly yes, but we are not the big windows gurus and haven't ever tried to set up a Windows box with an Apache(<http://www.apache.org>), Perl(<http://www.perl.com>) and MySQL(<http://www.mysql.com>). In case you tried it send us an e-mail.

23. How can I become a part of the OTRS developer community?

Welcome! Anybody who is willing to help us and has the time is more than welcome. Please send us an e-mail.

24. I do like the OTRS but would feel more comfortable by using a commercial product.

We can not help you. Sorry.

25. What browser do I need?

OTRS is working with the most browser such lynx, w3m, Netscape, Mozilla, Opera, IE and Konqueror (You don't need Java Script or Java Applets!).

26. Is it possible to generate my own ticket number format?

Yes it's possible. You can choose between four different OTRS formats. AutoIncrement ("SystemID.Counter" e. g. 1010138 or 1010139), Date ("Year.Month.Day.SystemID.Counter" e. g. 200206231010138 or 200206231010139), Random ("SystemID.Random" e. g. 100057866352 or 103745394596) and my favorite one DateChecksum ("Year.Month.Day.SystemID.Counter.CheckSum" e. g. 2002070110101520 and 2002070110101535).

Further it's possible to create a own ticket number format (like you want). See the OTRS documentation.

27. Do you support the RFC 1297?

Yes, OTRS supports this RFC.

28. Can I add my own ticket states?

Yes, but be carefully with the existing states (OTRS needs basic states).

29. More then one OTRS on one machine?

Yes, you can do that. Just install the second, third, ... in different paths (e.g. /opt/otrs01, /opt/otrs02, /opt/otrs03, ...). And configure the webserver like README.webserver.

Glossary

This glossary covers OTRS terms.

User Groups

Agent

The person who works on the tickets.

Admin

Each Agent who is (rw) in the 'admin' group is admin.

Customer

The problem reporter.

Web Interface

Agent Interface

Web Interface (<http://example.com/otrs/index.pl>) for the agent to work on the tickets.

Admin Interface

Web Interface (<http://example.com/otrs/index.pl?Action=Admin>) for the admin to do the admin jobs.

Customer Interface

Web Interface (<http://example.com/otrs/customer.pl>) for the customer to create, view and send follow ups for tickets.

Agent Interface

Mailbox

Overview of all agent locked tickets.

Phone View

A screen to create tickets.

Queue View

Overview of all 'open' and 'unlocked' tickets sorted by queue. Sort order of each queue is priority then age.

Ticket History

A detail view (each action, e. g. move, lock, ...) of a ticket.

Ticket Zoom

A detail view (each article) of a ticket.

You have n new message(s)!

New messages in agent locked tickets where the last article isn't written by the agent (e. g. by customer or other agent). The new message notify will go if last article is written by the agent (helps the agent to show what customer needs to be contacted).

Ticket

Ticket Number

Uniq ticket number. Reference for agent and customer. It's possible to change the ticket number format.

Ticket Age

Age of the ticket.

Ticket Status

Ticket states are e.g 'new' or 'open'. It's also possible to change/add/remove ticket states via the Admin Interface.

Ticket Priority

How important a ticket is, ticket priorities are e.g '3 normal' or '4 high'. It's also possible to change/add/remove ticket priorities (see doc).

Ticket Lock

A Ticket is 'locked' or 'unlocked'. That means it's locked for an agent who want's to work on it. At this time it's not writable for other agents! If an agent locks a ticket then the owner will be updated to the current agent automatically.

Ticket Owner

The current owner of the ticket. Owner isn't equal Lock!

Ticket CustomerID

The CustomerID of the customer. Default is the email address.

Ticket CustomerUser

The CustomerUser (the sender). Default is the email address.

Ticket Escalation Time

If an escalation time is set for a queue (Admin Interface -> Queue) and a ticket is open and not answered then just this ticket will be shown. No other ticket is accessible via QueueView.

Article

The parts of a ticket (requests, notes, answers) are articles.

Article Type

The article types:

email-external: email between customer and agent

email-internal: email between agents e. g. for feedback (_not_ shown in customer interface)

email-notification-ext: email notification to customer

email-notification-int: email notification to agent (_not_ shown in customer interface)

phone: phone note

webrequest: request from the customer web interface

note-internal: note just shown for agents (agent to agent - _not_ shown in customer interface)

note-external: note shown for agent and customer (should _not_ be used as communication between customer and agent!)

note-report (note used for reporting - should be used by monitoring tools like Nagios (<http://www.nagios.org/>))

[not used at this moment: fax, sms]

Article Sender Type

The creator of the article ('agent', 'customer' or 'system').

Response

Modular default answers for Agent Interface (to write faster answers). Configurable per queue.

Auto-Response

Automatically generated responses after customer created a new ticket or wrote a follow up. Configurable per queue.

Appendix A. Online resources

We try to support you with the very last information about OTRS and give you a good way to provide us with your feedback.

A.1. Homepage

Our homepage can be found at <http://www.otrs.org/>.



A.2. Mailinglists

We provide five major mailinglists. You must be subscribed to post to any of them.

The first is <announce at otrs.org>. It is a low traffic list for announcements of new OTRS releases and security issues.

The second mailinglist is <otrs at otrs.org>. It is a medium to high traffic list with all sorts of questions and support to the product.

The third mailinglist is <dev at otrs.org>. It is a medium to high traffic list. The OTRS developers discuss various design and implementation issues here.

The fourth mailinglist is <i18n at otrs.org>. It is a low traffic list for internationalization and localization questions. If you are or want to become a translator of the OTRS project or have any problems with one of our applications in an international environment, this is the right place.

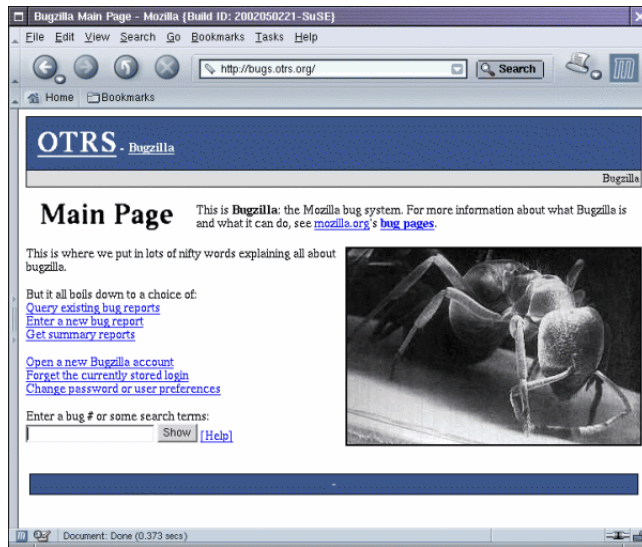
The fifth mailinglist is <cvcs-log at otrs.org>. It is a very high traffic list. CVS commits notifications.

To subscribe visit <http://lists.otrs.org/>.

A.3. Bugtracking

Real geeks don't need a bugtracking tool! ;-)

Just kidding. To submit bugs visit <http://bugs.otrs.org/>. We know sometimes bugzilla isn't that comfortable but right know it is the best bug tracking system we've found.



By reporting bugs you do help us very much. We appreciate your help!

Appendix B. The OTRS core team

OTRS was created in the third or fourth quarter of the year 2001 (nobody remembers the exact date). In those glory days the core team consisted of Martin Edenhofer and Stefan Wintermeyer.

B.1. Martin Edenhofer

E-Mail: [<martin@otrs.org>](mailto:martin@otrs.org)
Homepage: <http://martin.edenhofer.de/>

Without Martin the otrs wouldn't exist. He is a brilliant coder. We will insert some more information about Martin sometime. He is just to shy. ;-)

B.2. Stefan Wintermeyer

E-Mail: [<stefan@otrs.org>](mailto:stefan@otrs.org)
Homepage: <http://www.wintermeyer.de/>

Here is his "official" vita (German version only):

Stefan Wintermeyer arbeitet seit dem ersten ZX81 Selbstbausatz mit Computern. 1994 kam seine Leidenschaft für Linux dazu, die ihn schließlich 1998 zur SuSE Linux AG brachte. Als Vice President Support der SuSE Linux AG setzte er sich für eine größere Verbreitung und Benutzerfreundlichkeit von Linux ein und unterstützt konsequent echte Linuxlösungen im unternehmerischen Einsatz. Aus seinem Engagement bei der SuSE Linux AG blieb ihm die Vorliebe für die gleichnamige Distribution. Seit Juni 2001 arbeitet er bei der Lufthansa Systems in Frankfurt und betreut dort den Einsatz von Linux in verschiedenen Projekten. Stefan Wintermeyer setzt sich nachhaltig für die Verwendung offener Standards ein und propagiert Linux als das Betriebssystem für Server und Thin Clients.

B.3. Sebastian Wormser

E-Mail: [<sibbi@sibbi.com>](mailto:sibbi@sibbi.com)
Homepage: <http://sibbi.org/>
joined the team: 01.05.2002

We are very happy that Sebastian Wormser (Sibbi) finally joined our team. We have worked on a commercial trouble ticket system in the past (the stts at SuSE(<http://www.suse.de/>) - IBM Case Study on SuSE(<http://www.ibm.com/software/success/cssdb.nsf/CS/NAVO-56G3KU?OpenDocument>) / German Linux magazin(<http://www.linux-magazin.de/ausgabe/2001/04/stts/stts.html>)). So it is sort of a reunion but for an open source and total new (probably better) version.

Appendix C. Credits

As most open source projects we have to thank many people for there help. This is a non-complete list of those folks:

Martin Scherbaum, Carsten Gross, Harald Müller, Stefan Schmidt, Milisav Radmanic, Uli Hecht, Norman Walsh, Heiko Baumann, Atif Ghaffar, Pablo Ruiz Garcia, Dan Rau, Christoph Kaulich, Mark Jackson, Diane Shieh, Bernard Choppy, Carl Bailey, Phil Davis, Edwin D. Vinas, Lars Müller, Bryan Fullerton, Vladimir Gerdjikov, Fred van Dijk, Sebastien Guilbaud, Wiktor Wodecki, Arnold Ligtoet, Antti Kämäräinen, Nicolas Goralski, Robert Kehl, Gilberto Cezar de Almeida, Jorge Becerra, Eddie Urenda, Stella Power, Andreas Haase, Reiner Keller, Covert Jake, Moshe Leibovitch, Bjoern Jacke, Remo Catelotti;

And of course there are allways people who give a very special extra support to the project. Here they are:

- Robert Kehl, who has created the win32 installation section by himself. Thank you Robert!
- The OTRS mailinglists are a great source of ideas and give everybody very good support. Thank you guys for helping us!

mirrors

The project pretty soon reached a point where our ftp server didn't have enough bandwidth to serve the demand. Here is a list of people who rescued us by setting up mirrors. We appreciate your help!

Nils Jeppe (mirror Hamburg), Bryan Fullerton (ftp.samurai.com), Eberhard Moenkeberg (ftp.gwdg.de)

Appendix D. RFC 1297

Many people do not have an idea what a trouble ticket system is and why you may need one. The *RFC 1297* (<http://www.faqs.org/rfcs/rfc1297.html>) is a good start to get an overview.

RFC 1297

PURPOSES OF A NOC TROUBLE TICKET SYSTEM

A good Network Operations Trouble Ticket System should serve many purposes:

1) **SHORT-TERM MEMORY AND COMMUNICATION ("Hospital Chart").** The primary purpose of the trouble ticket system is to act as short-term memory about specific problems for the NOC as a whole. In a multi-operator or multi-shift NOC, calls and problem updates come in without regard to who worked last on a particular problem. Problems extend over shifts, and problems may be addressed by several different operators on the same shift. The trouble ticket (like a hospital chart) provides a complete history of the problem, so that any operator can come up to speed on a problem and take the next appropriate step without having to consult with other operators who are working on something else, or have gone home, or are on vacation. In single-room NOCs, an operator may ask out loud if someone else knows about or is working on a problem, but the system should allow for more formal communication as well.

2) **SCHEDULING and WORK ASSIGNMENT.** NOCs typically work with many simultaneous problems with different priorities. An on-line trouble ticket system can provide real time (or even constantly displayed and updated) lists of open problems, sorted by priority. This would allow operators to sort their work at the beginning of a shift, and to pick their next task during the shift. It also would allow supervisors and operators to keep track of the current NOC workload, and to call in and assign additional staff as appropriate.

It may be useful to allow current priorities of tickets change according to time of day, or in response to timer alerts.

3) **REFERRALS AND DISPATCHING.** If the trouble ticket system is thoroughly enough integrated with a mail system, or if the system is used by Network Engineers as well as Network Operators, then some problems can be dispatched simply by placing the appropriate Engineer or Operator name in an "assigned to" field of the trouble ticket.

4) **ALARM CLOCK.** Typically, most of the time a trouble ticket is open, it is waiting for something to happen. There should almost always be a timer associated with every wait. If a ticket is referred to a phone company, there will be an escalation time before which the phone company is supposed to call back with an update on the problem. For tickets referred to remote site personnel, there may be other more arbitrary timeouts such as

"Monday morning". Tickets referred to local engineers or programmers should also have timeouts ("Check in a couple of days if you don't hear back from me"). A good trouble ticket system will allow a timeout to be set for each ticket. This alarm will generate an alert for that ticket at the appropriate time. Preferably, the system should allow text to be attached to that timer with a shorthand message about what the alert involves ("Remind Site: TT xxx") (The full story can always be found by checking the trouble ticket). These alerts should feed into the NOC's standard alert system.

The Alarm Clock can also assist (or enforce!) administrative escalation. An escalation timer could automatically be set based on the type of network, severity of the problem, and the time the outage occurred.

5) **OVERSIGHT BY ENGINEERS AND CUSTOMER/SITE REPRESENTATIVES.** NOCs frequently operate more than one network, or at least have people (engineers, customer representatives, etc) who are responsible for subsets of the total network. For these individual representatives, summaries of trouble tickets can be filtered by network or by node, and delivered electronically to the various engineers or site representatives. Each of these reports includes a summary of the previous day's trouble tickets for those sites, a listing of older trouble tickets still open, and a section listing recurrent problems. These reports allow the site reps to keep aware the current outages and trends for their particular sites. The trouble ticket system also allows network access to the details of individual trouble tickets, so those receiving the general reports can get more detail on any of their problems by referencing the trouble ticket number.

6) **STATISTICAL ANALYSIS.** The fixed-form fields of trouble tickets allow categorizations of tickets, which are useful for analyzing equipment and NOC performance. These include, Mean Time Between Failure and Mean Time to Repair reports for specific equipment. The fields may also be of use for generating statistical quality control reports, which allow deteriorating equipment to be detected and serviced before it fails completely. Ticket breakdowns by network a NOC costs to be apportioned appropriately, and help in developing staffing and funding models. A good trouble ticket system should make this statistical information in a format suitable for spreadsheets and graphics programs.

7) **FILTERING CURRENT ALERTS.** It would be possible to use network status information from the trouble ticket system to filter the alerts that are displayed on the alert system. For instance, if node XXX is known to be down because the trouble ticket is currently open on it, the alert display for that node could automatically be acknowledged. Trouble tickets could potentially contain much further information useful for expert system analysis of current network alert information.

8) **ACCOUNTABILITY ("CYA"), FACILITATING CUSTOMER FOLLOW-THROUGH, AND NOC IMAGE).** Keeping user-complaint tickets facilitates the kind of follow through with end-users that generates happy clients (and good NOC image) for normal trouble-fixing situations. But also, by their nature, NOCs deal with crises; they occasionally find themselves with major outages, and angry users or administrators. The trouble ticket system documents the NOC's (and the rest of the organization's) efforts to solve problems in case of complaints.

Of course we added many features to the OTRS which are not mentioned in this RFC. And we will add many features.

Anyhow we are keen on your feedback. Please do not hesitate to send us an e-mail to <feedback@otrs.org>

Your OTRS core team

Appendix E. GNU Free Documentation License

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies,

and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Bibliography

Ben Laurie and Peter Laurie, 1999, O'Reilly, *Apache. Das umfassende Referenzwerk.*, 3897211270.

Bryan Costales and Eric Allman, 1997, O'Reilly, *sendmail*, 1565922220.

Richard Blum, 2001, SAMS, *Postfix*, 0672321149.

Eric Rescorla, 2000, Addison Wesley, *SSL and TLS*, 0201615983.

iX, Heise Verlage, "Winnetou und Old SSLeay", Holger Reif, 1998, page 128.

George Reese, Tim King, and Randy Jay Yarger, 2002, O'Reilly, *Managing & Using MySQL (2nd Edition)*, 0596002114.

Illiad, 1999, O'Reilly, *User Friendly*, 1565926730.