

OTRS 2.0 - Admin Manual

OTRS 2.0 - Admin Manual

Klong Jark Beach Edition

Copyright © 2003-2006 OTRS GmbH

Richard Kammermeyer, Christian Schöpplein, Stefan Rother, Thomas Raith, Burchard Steinbild, Andre Mindermann, Christopher Kuhn, Martin Edenhofer

This work is copyrighted by OTRS GmbH.

You may copy it in whole or in part as long as the copies retain this copyright statement.

UNIX is a registered trademark of X/Open Company Limited. Linux is a registered trademark of Linus Torvalds.

MS-DOS, Windows, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP and Windows 2003 are registered trademarks of Microsoft Corporation. Other trademarks and registered trademarks are: SUSE and YaST of SUSE GmbH, Red Hat and Fedora are registered trademarks of Red Hat, Inc. Mandrake is a registered trademark of MandrakeSoft, SA. Debian is a registered trademark of Software in the Public Interest, Inc.

All trade names are used without the guarantee for their free use and are possibly registered trade marks.

OTRS GmbH essentially follows the notations of the manufacturers. Other products mentioned in this manual may be trademarks of the respective manufacturer.

Table of Contents

Preface	ix
1. Basics about trouble ticket systems.....	1
1.1. What is a trouble ticket system, when do you need it?	1
1.2. What is a trouble ticket?.....	2
2. OTRS - Open Ticket Request System	3
2.1. Basics	3
2.2. Features	3
2.3. Hard and software requirements	5
2.4. Community.....	6
2.5. Commercial Support for OTRS.....	7
3. Installation of the OTRS framework	8
3.1. The simple way - Installation of prebuilt packages.....	8
3.1.1. Installing the rpm on a SUSE distro	8
3.1.2. Installing OTRS on a Debian system	14
3.1.3. Installing OTRS on Microsoft Windows systems	14
3.2. Installation from source (Linux, Unix)	14
3.2.1. Preparing the installation from source.....	15
3.2.2. Installation of needed perl modules.....	16
3.2.3. Configuring the apache web server	18
3.2.4. Configuring the database	19
3.2.5. Setting up the cron jobs for OTRS	24
4. First steps in OTRS.....	29
4.1. Agent web interface	29
4.2. Customer web interface.....	30
4.3. Public web interface.....	30
4.4. First login	31
4.5. An overview to the web interface.....	32
4.6. What is a queue?	35
4.7. User preferences.....	36
5. The admin area of OTRS	39
5.1. Basics	39
5.2. User, groups and roles	40
5.2.1. User.....	40
5.2.2. Groups	41
5.2.3. Roles	43
5.3. Customer users and customer groups.....	46
5.3.1. Customer users	46
5.3.2. Customer groups.....	47
5.4. Queues.....	48
5.5. Salutations, signatures, attachments and responses	50
5.5.1. Salutations	52
5.5.2. Signatures	53
5.5.3. Attachments.....	54
5.6. Auto answers.....	56

5.7. Email addresses	58
5.8. Notifications	59
5.9. SMIME.....	60
5.10. PGP	61
5.11. States	62
5.12. The graphical configuration frontend (SysConfig)	63
5.13. Using POP3 accounts.....	64
5.14. Filtering incoming messages.....	65
5.15. Executing automated jobs with the GenericAgent.....	68
5.16. Admin email.....	69
5.17. Session management	70
5.18. System Log	71
5.19. SQL queries via the select box.....	72
5.20. Package manager.....	73
6. Configuring the system.....	75
6.1. The config files of OTRS	75
6.2. Configuring the system through the web interface	75
7. Sending/Receiving emails.....	78
7.1. Sending emails	78
7.1.1. Via Sendmail (default).....	78
7.1.2. Via SMTP relay/smarthost	78
7.2. Receiving emails	78
7.2.1. Via POP3 accounts - the easy way (PostMasterPOP3.pl)	79
7.2.2. Via command line program and e.g. procmail (PostMaster.pl).....	80
7.2.3. Fetching emails via POP3 or IMAP and fetchmail for PostMaster.pl.....	81
7.2.4. Filtering/dispatching by OTRS/PostMaster modules (for more complex dispatching)81	
8. Time related functions in OTRS.....	84
8.1. Specifying the relevant times for OTRS	84
8.1.1. TimeWorkingHours	84
8.1.2. TimeVacationDays.....	84
8.1.3. TimeVacationDaysOneTime.....	85
8.2. Automated unlocking of tickets	86
8.3. Pending tickets	86
8.4. Ticket escalation.....	86
9. Using external backends.....	88
9.1. Customer data	88
9.2. Customer user backend	88
9.2.1. Database (Default).....	88
9.2.2. LDAP.....	92
9.2.3. Use more than one customer backend with OTRS	94
9.3. Backends to authenticate agents and customer users.....	97
9.3.1. Authentication backends for agents	97
9.3.2. Authentication backends for customer users.....	100
9.4. Customize the customer self registration	103
9.4.1. Customizing the web interface	103
9.4.2. Customer mapping.....	103

9.4.3. Customize the customer_user table in the OTRS DB	105
10. Modifying ticket states and ticket state types.....	106
11. Modifying ticket priorities	109
12. Creating own themes	111
13. Localization of the OTRS frontend	112
14. PGP	113
15. S/MIME	117
16. Additional applications	121
16.1. Calendar	121
16.2. Content manager	121
16.3. File manager.....	122
16.4. Web mailer	122
16.5. FAQ	123
16.6. System status	123
17. Performance Tuning	125
17.1. OTRS.....	125
17.1.1. TicketIndexModule.....	125
17.1.2. TicketStorageModule	125
17.2. Database	126
17.2.1. MySQL	126
17.2.2. PostgreSQL.....	127
17.3. Webserver.....	127
17.3.1. Preestablish datababase connections	127
17.3.2. Preloaded modules - startup.pl	127
17.3.3. Reload perl modules when updated on disk	127
17.3.4. Choosing the Right Strategy	128
17.3.5. mod_gzip/mod_deflate	128
17.3.6. mod_dosevasive.....	128
18. Backing up the system.....	129
18.1. Backup	129
18.2. Restore	130
A. Additional Resources	131
A.1. Homepage OTRS.org	131
A.2. Mailinglists.....	131
A.3. Bugtracking.....	132
A.4. Commercial Support	133
B. Config Referenzliste	134
B.1. Framework.....	134
B.1.1. Core.....	134
B.1.2. Core::Log	141
B.1.3. Core::MIME-Viewer.....	143
B.1.4. Core::Package	144
B.1.5. Core::Sendmail	145
B.1.6. Core::Session	148
B.1.7. Core::SpellChecker.....	152

B.1.8. Core::Time	153
B.1.9. Core::Web	156
B.1.10. Crypt::PGP	158
B.1.11. Crypt::SMIME	160
B.1.12. Frontend::Admin::ModuleRegistration	161
B.1.13. Frontend::Agent	171
B.1.14. Frontend::Agent::Auth::LDAP	174
B.1.15. Frontend::Agent::ModuleNotify	175
B.1.16. Frontend::Agent::ModuleRegistration	176
B.1.17. Frontend::Agent::Preferences	180
B.1.18. Frontend::Customer	184
B.1.19. Frontend::Customer::Auth	189
B.1.20. Frontend::Customer::ModuleNotify	195
B.1.21. Frontend::Customer::ModuleRegistration	196
B.1.22. Frontend::Customer::Preferences	198
B.2. Ticket	202
B.2.1. Core::LinkObject	202
B.2.2. Core::PostMaster	202
B.2.3. Core::Stats	210
B.2.4. Core::Ticket	212
B.2.5. Core::TicketFreeText	222
B.2.6. Frontend::Admin::ModuleRegistration	231
B.2.7. Frontend::Agent	238
B.2.8. Frontend::Agent::ModuleNotify	247
B.2.9. Frontend::Agent::ModuleRegistration	247
B.2.10. Frontend::Agent::NavBarModule	259
B.2.11. Frontend::Agent::Preferences	259
B.2.12. Frontend::Agent::Ticket::ArticleAttachmentModule	264
B.2.13. Frontend::Agent::Ticket::ArticleComposeModule	265
B.2.14. Frontend::Agent::Ticket::ArticleViewModule	265
B.2.15. Frontend::Agent::Ticket::ArticleViewModulePre	266
B.2.16. Frontend::Agent::Ticket::MenuModule	267
B.2.17. Frontend::Agent::Ticket::MenuModulePre	272
B.2.18. Frontend::Agent::Ticket::ViewBounce	275
B.2.19. Frontend::Agent::Ticket::ViewClose	276
B.2.20. Frontend::Agent::Ticket::ViewCompose	278
B.2.21. Frontend::Agent::Ticket::ViewEmailNew	279
B.2.22. Frontend::Agent::Ticket::ViewForward	282
B.2.23. Frontend::Agent::Ticket::ViewHistory	283
B.2.24. Frontend::Agent::Ticket::ViewMailbox	283
B.2.25. Frontend::Agent::Ticket::ViewMerge	284
B.2.26. Frontend::Agent::Ticket::ViewMove	285
B.2.27. Frontend::Agent::Ticket::ViewNote	285
B.2.28. Frontend::Agent::Ticket::ViewOwner	288
B.2.29. Frontend::Agent::Ticket::ViewPending	289
B.2.30. Frontend::Agent::Ticket::ViewPhone	291
B.2.31. Frontend::Agent::Ticket::ViewPhoneNew	293
B.2.32. Frontend::Agent::Ticket::ViewQueue	296

B.2.33. Frontend::Agent::Ticket::ViewSearch	299
B.2.34. Frontend::Agent::Ticket::ViewStatus	303
B.2.35. Frontend::Agent::Ticket::ViewZoom.....	304
B.2.36. Frontend::Customer	305
B.2.37. Frontend::Customer::ModuleRegistration	314
B.2.38. Frontend::Customer::Preferences	317
B.3. FAQ.....	318
B.3.1. Core.....	319
B.3.2. Core::LinkObject	321
B.3.3. Frontend::Agent::ModuleRegistration.....	321
B.3.4. Frontend::Customer::ModuleRegistration	324
B.3.5. Frontend::Public.....	324
B.3.6. Frontend::Public::ModuleRegistration	325
C. Credits	326
D. GNU Free Documentation License.....	328
0. PREAMBLE	328
1. APPLICABILITY AND DEFINITIONS	328
2. VERBATIM COPYING.....	329
3. COPYING IN QUANTITY	329
4. MODIFICATIONS.....	330
5. COMBINING DOCUMENTS.....	331
6. COLLECTIONS OF DOCUMENTS	332
7. AGGREGATION WITH INDEPENDENT WORKS.....	332
8. TRANSLATION	333
9. TERMINATION.....	333
10. FUTURE REVISIONS OF THIS LICENSE.....	333
How to use this License for your documents	333

List of Tables

3-1. Needed perl modules for OTRS	16
3-2. Description of the several cron job scripts	25
5-1. Default groups after OTRS has been installed	41
5-2. Rights in the user groups of OTRS.....	43
5-3. Events for auto answers.....	57
5-4. Function of the different X-OTRS-headers	66
A-1. Mailinglists.....	131

Preface

This book wants to make life easier for OTRS administrators and OTRS newbies. The following chapters describe the installation, configuration and administration of the ticket system, making this book less interesting for agents or customer users.

Although many hours of work, some litres of coffee and several pizzas were invested into the following sections, it is still not complete. Inevitably there will be errors, parts with complicated explanations or missing chapters about important things. Some chapters will be reworked or new chapters and sections will be added.

Since the book tries to satisfy the needs of OTRS admins or OTRS newbies and since the quality should be as good as possible, we need your feedback. Please write to us if you find something missing in this book, if things are not explained well enough or even if you see spelling mistakes, grammatical errors or typos. Any kind of feedback is highly appreciated and should be made via our bug tracking system on <http://bugs.otrs.org> (<http://bugs.otrs.org>) . We want to thank you for all kinds of feedback!

Chapter 1. Basics about trouble ticket systems

This chapter gives a short overview about the idea of trouble tickets in general and trouble ticket systems in detail. A short example shows the advantages of a ticket system.

1.1. What is a trouble ticket system, when do you need it?

The following example describes what a trouble ticket system is and how you can save time and money if you use such a system in your company.

Let's imagine that Max is a manufacturer who produces video recorders. Since programming video recorders is very complex, Max gets many mails from customers who need help with the devices. Some days, Max has not enough time to answer all mails so that his customers have to be patient until they get an answer. Some customers don't want to wait that long so that they write another mail with the same question. All mails containing support requests are stored in only one inbox file. The requests are not sorted and the mails are answered with a normal mail program.

Then the developers Joe and John have to help Max answer the mails because Max can't reply fast enough to all the messages. Since Max, Joe and John use the same system with the same inbox file, they all access the same inbox. Joe and John don't know that Max often got two identical requests from a desperate customer. So Joe answers the first and John answers the second mail of the customer, both with a helpful hint on how to solve the problem. The customer receives two different answers, and Max does not know what John and Joe wrote. Max neither has an overview about the replies of Joe and John nor does he know all the problems the customers have had with the video recorders nor which problems occur very often nor how much time and money he has to spend for customer support.

At a meeting, a colleague tells Max about trouble ticket systems and that they can solve Max's problems with the customer support. After looking for information in the internet, Max decides to install the Open Ticket Request System (OTRS) on a computer that can be reached via the web by both his customers and his employees. Now, the customer requests are no longer sent to Max's private inbox but to the mail account that is used for OTRS. The ticket system is connected to this mailbox and saves all requests in its database. For every new request, the system generates an auto-answer and sends it to the customer so that the customer knows that his request has arrived and will be answered soon. OTRS generates an explicit identification, the ticket number, for every single request. Now, the customers are happy because they got a quick response to their requests and it is not necessary to send a second message with the same question. Max, John and Joe can now login into OTRS with a simple web browser and answer the requests. Since the system locks a ticket which is answered, no message is edited twice.

Let's imagine that Mr Smith makes a request to Max's company and his message is processed by OTRS. John gives a brief reply to this request but Mr Smith wants to know more so he replies to John's mail.

Since John has other things to do, Max answers Mr Smith's message. The history function of OTRS makes it easy for Max to read both Mr Smith's first message and John's response so that he is able to write a more detailed reply to Mr Smith. Mr Smith does not know that two different people were involved in the communication process, he is only happy about the solution for his problem that arrived with Max's last reply.

Of course, this is only a short overview about the possibilities and features of trouble ticket systems. But if many customer requests (mails and phone calls) must be handled by different people, a ticket system can help to save working time and money. A ticket system helps you to structure your support or helpdesk environment and you can define workflow processes. The communication between customers and companies gets more transparent which results in more effectiveness for both sides.

1.2. What is a trouble ticket?

A trouble ticket is comparable to a medical report for a hospital patient. A medical report is generated when the patient visits the hospital for the first time. All important data about the patient, such as personal data and facts about the medical problem, are noted in the medical report. Each doctor adds to the medical report information on the illness and any medication so that all doctors and the whole nursing staff can get a quick and detailed overview about the patient's problem. When the patient has recovered and leaves hospital, the medical report gets closed and all information get archived.

Trouble ticket systems like OTRS, handle trouble tickets like normal email. The messages are saved in the system. When a customer sends a request, a new ticket is generated by the system which is comparable to a new medical report being opened. The answer to this new ticket can be compared to a doctor's entry in the medical report, a new answer to a change in the clinical picture of the patient. A ticket is finished if an answer is send back to the customer or if the ticket is closed by the system. If a customer sends an answer for a already closed ticket, the ticket will be reopened and the new information will be added. To keep the consistency of all data in the system, every ticket is stored and archived with all relevant information. Since tickets are handled like normal emails, attachments will be stored, too. Notes can be added as additional information to every email. All tickets are stored on the hard drive or in the database of the system. Also additional information like notes, dates, employees involved, working time needed for a ticket etc. are stored. All tickets can be sorted and it is possible to search through all data.

Chapter 2. OTRS - Open Ticket Request System

This chapter describes the features of the Open Ticket Request System (OTRS). You will find informations about the hardware and software requirements for OTRS. Additionally, this chapter tells you how to get commercial support for OTRS, should you require it, and how to contact the community.

2.1. Basics

The Open Ticket Request System (OTRS) is a web application which can be used with every HTML-compatible web browser. The web interface of OTRS does not use active web content like Flash or Java applets to ensure that the system is usable with mobile phones or other mobile computers. To use OTRS, no special client operating system is necessary; only an HTML browser is needed.

OTRS is separated into several components. The basic component is the OTRS framework that contains all central functions for the application and the ticket system. Via the web interface of the central OTRS framework, it is possible to install additional applications like a web mailer, a content manager, a file manager, a web calendar and a tool to monitor system status information.

2.2. Features

OTRS has many features. The following list gives an overview of the features included in the central framework.

The features of OTRS

- Web interface:
 - Easy and initial handling with a web browser.
 - Because no active web contents like Flash or Java applets are used, the web interface is usable with most web browsers, even with mobile phones or other mobile computers.
 - A web interface to administer the system via the web is available.
 - A web interface to handle customer requests by employees/agents via the web is integrated.
 - A web interface for customers is available to write new tickets, check the state and answer old tickets and search through their own tickets.
 - The web interface can be customized with different themes; own themes can be integrated.
 - Support for many languages.
 - The appearance of output templates can be customized (dtl).
 - Mails from and into the system can contain multiple attachments.

- Mail interface:
 - Support for mail attachments (MIME support).
 - Automatic conversion of HTML into plain text messages (more security for dangerous content and enables faster searching).
 - Mail can be filtered with the X-OTRS headers of the system or via mail addresses, e.g. for spam messages.
 - PGP support, creation and import of own keys, signing and encrypting outgoing mail, signed and encrypted messages can be displayed.
 - Support for viewing and encrypting SMIME messages, handling of SMIME certificates.
 - Auto answers for customers, configurable for every queue.
 - Email notifications for agents about new tickets, follow-ups or unlocked tickets.
 - Follow-ups by references or In-Reply-To header entries.
- Tickets:
 - Expanded queue view, fast overview of new requests in a queue.
 - Tickets can be locked.
 - Creation of own auto answer templates.
 - Creation of own auto responders, configurable for every queue.
 - Ticket history, overview of all events for a ticket (changes of ticket states, replies, notes, etc.).
 - Print view for tickets.
 - Adding own (internal or external) notes to a ticket (text and attachments).
 - Ticket zooming.
 - Access control lists for tickets can be defined.
 - Forwarding or bouncing tickets to other mail addresses.
 - Moving tickets between queues.
 - Changing/setting the priority of a ticket.
 - The working time for every ticket can be counted.
 - Up-coming tasks for a ticket can be defined (pending features).
 - Bulk actions on tickets are possible.
 - Automatic and timed actions on tickets are possible with the "GenericAgent".
 - Full text search on all tickets is possible.
- System:
 - OTRS runs on many operating systems (Linux, Solaris, AIX, FreeBSD, OpenBSD, Mac OS 10.x, Microsoft Windows).

- ASP support (active service providing).
- Linking several objects is possible, e.g. tickets and FAQ entries.
- Integration of external back-ends for the customer data, e.g. via AD, eDirectory or OpenLDAP.
- Setting up an own ticket identifier, e.g. Cal#, Ticket#, Request#, e.g.
- The integration of your own ticket counter is possible.
- Support of several database systems for the central OTRS back-end, e.g. MySQL, PostgreSQL, SAPDB, Oracle).
- Framework to create stats.
- utf-8 support for the front- and back-end.
- Authentication for customers via database, LDAP, HTTPAuth or Radius.
- Support of user accounts, user groups and roles.
- Support of different access levels for several systems components or queues.
- Integration of standard answer texts.
- Support of sub queues.
- Different salutations and signatures can be defined for every queue.
- Email notifications for admins.
- Information on updates via mail or the web interface.
- Escalation for tickets.
- Support for different time zones.
- Simple integration of own add-ons or applications with the OTRS API.
- Simple creation of own front-ends, e.g. for X11, console.

2.3. Hard and software requirements

OTRS can be installed on many operating systems. OTRS runs not only on linux and on other unix derivates (e.g. OpenBSD or FreeBSD) but on all Microsoft Windows platforms too. OTRS has no excessive hardware requirements. We recommend using a machine with at least a 2 GHz Xeon or comparable CPU, 2 GB RAM and a 160 GB hard drive.

If you want to use OTRS, you need some other software components. The basic software requirements are a web and a database server as well as a working perl environment with some additional modules. The web server and perl has to be installed on the same machine as OTRS. The database back-end can be installed on the local or on an other host.

For the web server, we recommend using apache 1.3.x or apache 2.x. With this web server, you can use the mod_perl module which improves greatly the performance of OTRS. If you can't use apache, OTRS should run on any web server that can execute perl scripts.

For database back-ends, we recommend to use MySQL (3.1.x or higher) or PostgreSQL. However, all database servers that use SQL for their database language should be able to work with OTRS. If you use MySQL you have the advantage that you can configure your database and some system settings during the installation through a web front-end.

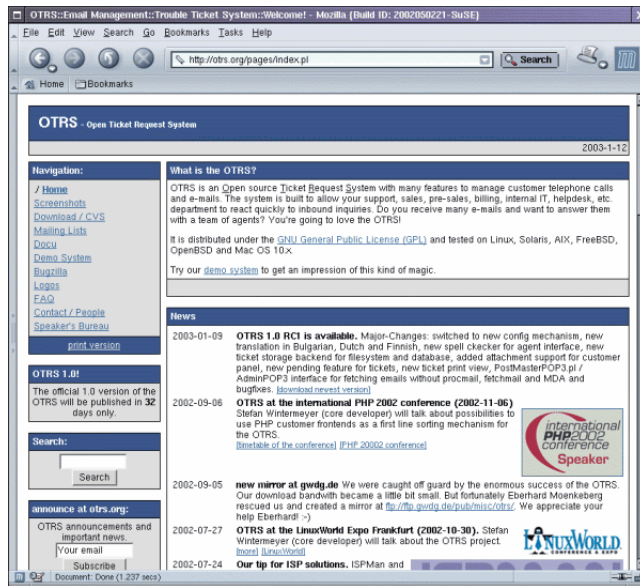
For perl, we recommend using at least version 5.8. You need some additional modules which can be installed either with the perl shell and CPAN or via the package manager of your operating system (rpm, yast, apt-get).

The section on the manual installation of perl modules needed describes in more detail how you can set up the perl modules that you need for OTRS.

If you install a binary package of OTRS, which was built for your operating system (rpm, Windows-Installer), either the package contains all perl modules needed or the package manager of your system should take care of the dependences of the perl modules needed.

2.4. Community

OTRS has a large user community. users and developers discuss OTRS and exchange information on the trouble ticket system Via mailinglists . There is available help for questions about the installation, configuration, usage, localisation and development. Software bugs can be reported via a bug tracking system so that they reach the responsible developers directly and without getting lost. That ensures that fixes for problems are available very quick.



Via the URL <http://www.otrs.org> (<http://www.otrs.org>) you can reach the homepage of the OTRS community.

2.5. Commercial Support for OTRS

Commercial support for OTRS is also available. Via the URL <http://www.otrs.com> (<http://www.otrs.com>) the pages of the OTRS GmbH can be reached, representing the commercial part of the OTRS.org project.

The OTRS GmbH provide support, consulting and training for the Open Ticket Request System. It also provides easy-to-handle complete solutions. With these packages, the OTRS GmbH would like to serve medium-sized companies, local authorities, institutions and big corporate groups. You can either buy ready-made high-performance and high-availability systems or order special made-to-measure packages.

On <http://www.otrs.com> (<http://www.otrs.com>) are available more detailed information about the OTRS GmbH. You can contact the OTRS GmbH per email to sales@otrs.com (<mailto:sales@otrs.com>)

Chapter 3. Installation of the OTRS framework

This chapter describes the installation and the basic configuration of the central OTRS framework. You'll find information about installing OTRS manually from source or with a binary package, ex. rpm or for Win32.

The configuration of the web and database server, the interface between OTRS and the database, the installation of needed perl modules, setting proper access rights for OTRS, setting up the cron jobs for OTRS and some basic settings in the config files of OTRS are described in this chapter.

When you have finished reading this chapter, you should have a running OTRS system installed on your operating system where you can login and administrate the system via the web interface.

3.1. The simple way - Installation of prebuilt packages

The simplest and most comfortable way to install OTRS is to use prebuilt packages. Many prebuilt packages of OTRS can be found in the download area on <http://www.otrs.org> (<http://www.otrs.org>). The following sections describe the installation of OTRS with a prebuilt or binary package on SUSE, Debian and Microsoft Windows systems. Use prebuilt packages to install OTRS and only setup OTRS manually, if you have no other possibility.

3.1.1. Installing the rpm on a SUSE distro

This section describes the installation of a prebuilt rpm package on a SUSE Linux distro. We tested the SUSE versions from 7.x to 10.0. Before you start the installation please have a look on <http://www.otrs.org> (<http://www.otrs.org/>) and check, if a newer OTRS rpm package is available. Please use the newer rpm package, if available.

Please install OTRS with yast (yast2) or via the command line and **rpm**. Because OTRS needs some perl modules which are not installed on a SUSE system by default, we recommend to use yast, because yast solves the package dependencies automatically.

If you decide to install OTRS via the command line and rpm, you have to install the needed perl modules manually before. Lets say, that you have saved the file `otrs.rpm` into the directory `/tmp`, you can execute the following command to install OTRS:

```
linux:~ # 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/>

linux:~ #

After the installation of the OTRS rpm package, you have to run SuSEconfig. Use the following command:

```
linux:~ # 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.
linux:~ #
```

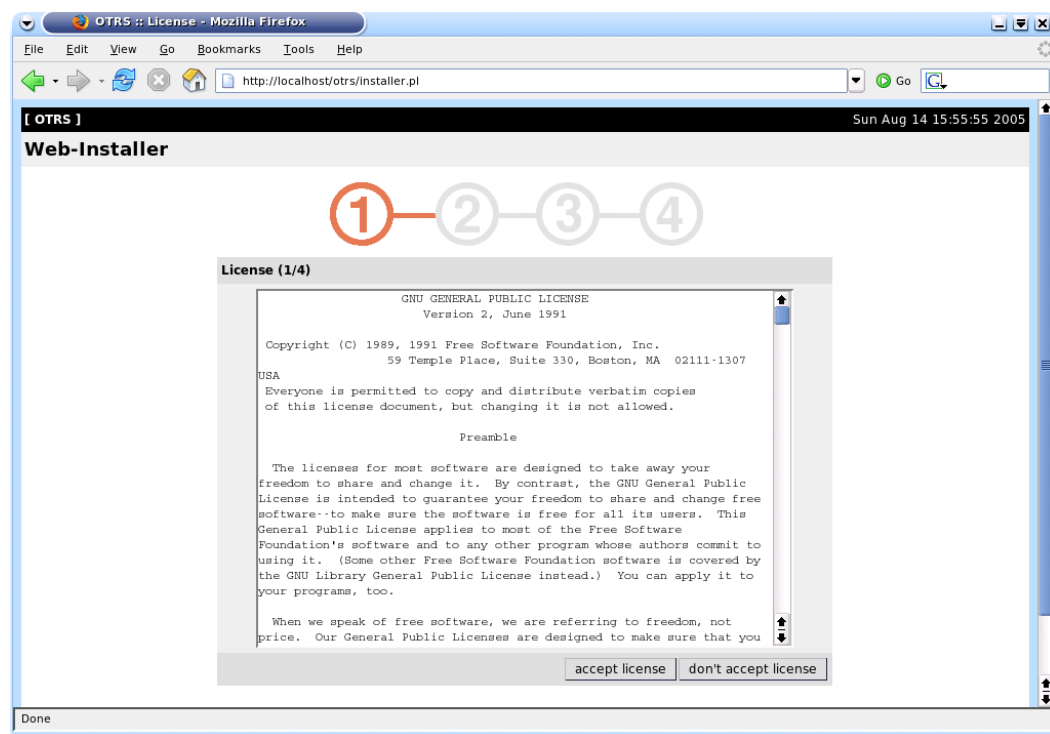
The installation of the OTRS rpm is finished. Restart your webserver to load the OTRS specific changes in your web server configuration:

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

In the next step you have to setup the OTRS database. If you use MySQL as the database backend, you can use the web installer of OTRS to setup the database. Use the following address to access the web installer start file.

<http://localhost/otrs/installer.pl> (<http://localhost/otrs/installer.pl>)

The web installer starts. Please follow the steps and setup the system.



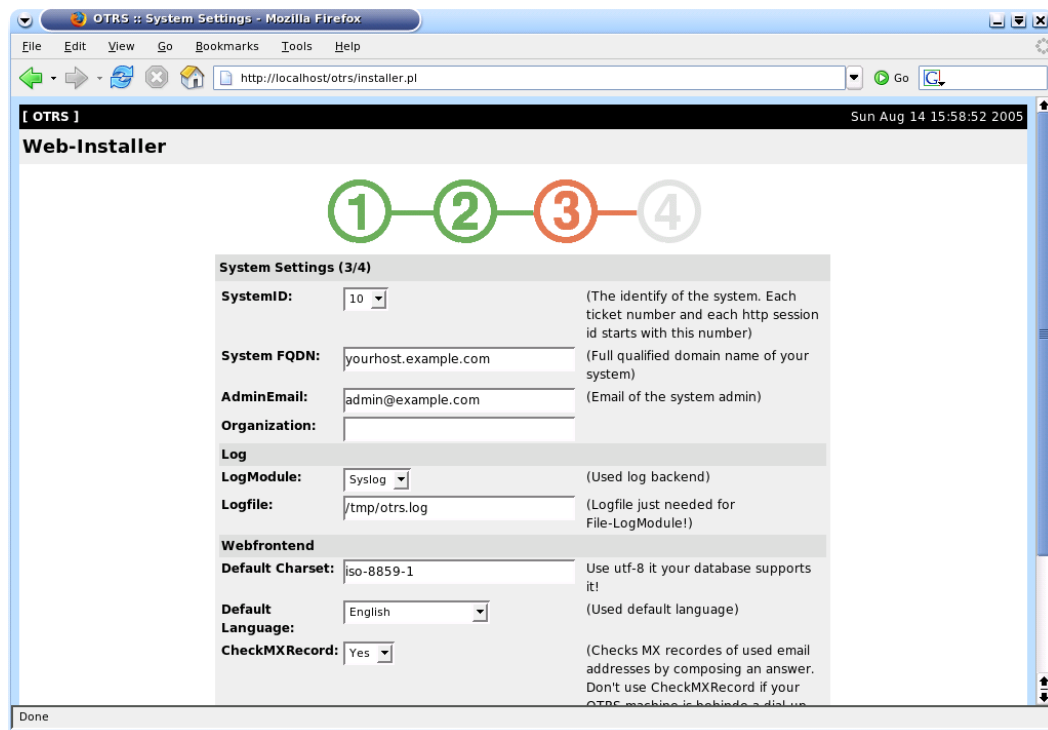
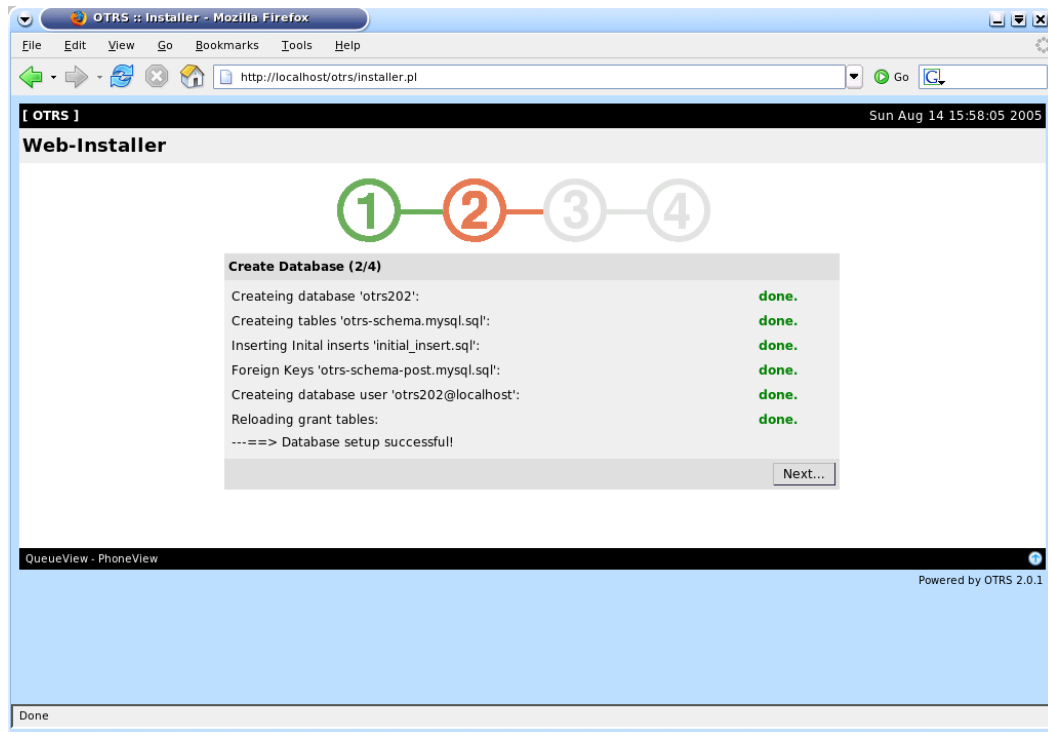
The screenshot shows the OTRS Web-Installer interface in a Mozilla Firefox browser window. The address bar shows `http://localhost/otrs/installer.pl`. The page title is "[OTRS]" and the timestamp is "Sun Aug 14 15:56:54 2005". The main heading is "Web-Installer". Above the form is a progress indicator with four numbered circles: 1 (green), 2 (red), 3 (grey), and 4 (grey). The form is titled "Create Database (2/4)". It contains the following fields and options:

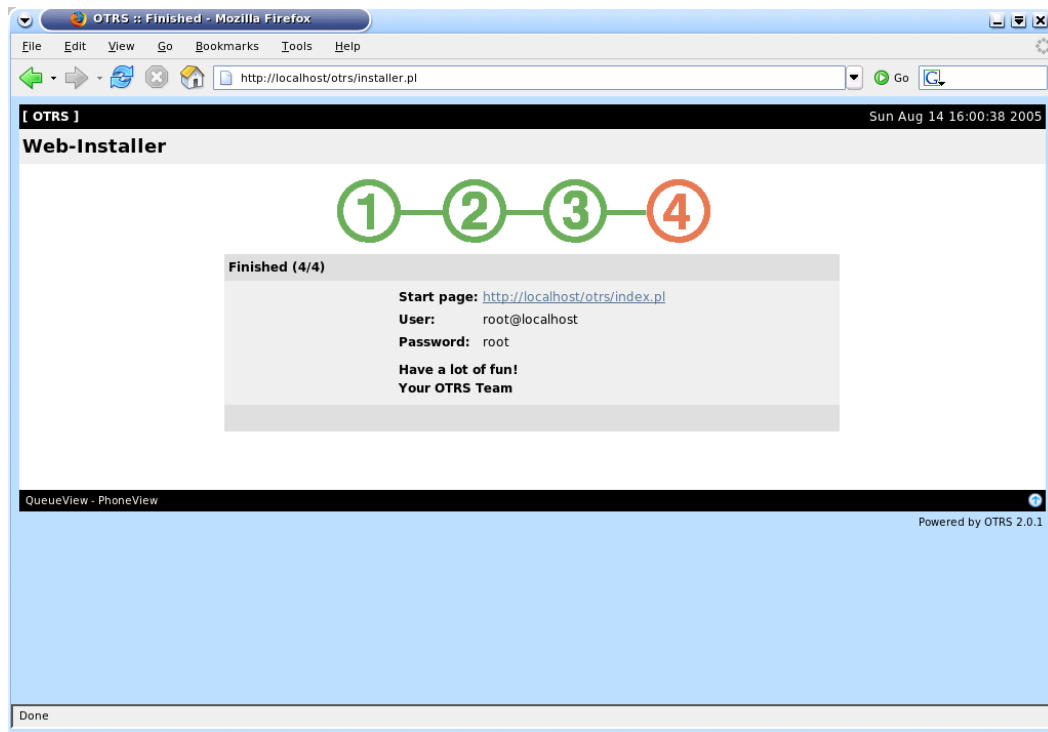
- Admin-User:**
- Admin-Password:**
- Host:**
- Type:**
- Database-User (New)**
 - User:**
 - Password:**
 - DB connect host:**
- Database**
 - Name:**
 - Action:** ☒ Create ☐ Delete

Red text annotations are present: "(* your MySQL DB should have a root password! Default is empty!)" next to the Admin-Password field, and "(default 'hot!)" next to the Database-User Password field. A "Next..." button is at the bottom right of the form. The browser's status bar at the bottom shows "QueueView - PhoneView" and "Done".

Warning

It is never a good idea to use default passwords. Please change the default password for the OTRS database!





Please restart the OTRS service now, to use the new configuration settings:

```
linux:~ # 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://linux.example.com/otrs/index.pl <--  
  
done  
done  
  
linux:~ #
```

Now the installation of OTRS is finished and you should be able to work with the system. To log into the web interface of OTRS, use the address <http://localhost/otrs/index.pl> (<http://localhost/otrs/index.pl>) in your web browser. Log in as OTRS admin and configure the system for your needs. To log in as OTRS admin use the username `root@localhost` and the password `root`.

Warning

Please change the password for the OTRS admin as soon as possible. It is also a default password!

3.1.2. Installing OTRS on a Debian system

Torsten Werner, the maintainer of the Debian OTRS package, kindly provided an excellent documentation on the installation of OTRS on Debian systems. The docu can be found on http://www.writely.com/View.aspx?docid=drm3kmx_0cbr3x9 (http://www.writely.com/View.aspx?docid=drm3kmx_0cbr3x9).

3.1.3. Installing OTRS on Microsoft Windows systems

Installing OTRS on a Microsoft Window system is very easy. Download the latest installer for Win32 from <http://www.otrs.org> (<http://www.otrs.org/>) and save the file to your local file system. Then simply click on the file to execute the installer. Follow the few installation steps to setup the system, after installation you should be able to login as OTRS administrator.

Important: The Win32 installer for OTRS contains all needed components for the trouble ticket system. That means, that also the apache2 web server, the MySQL database server, perl with all needed modules and cron for Windows will be installed. For that reason you should only install OTRS on Windows systems that don't already have apache2 or another web server and MySQL installed.

3.2. Installation from source (Linux, Unix)

3.2.1. Preparing the installation from source

If you have to install OTRS from source, first download the .tar.gz or .tar.bz2 file with the sources from <http://www.otrs.org> (<http://www.otrs.org/>) please.

Unpack the archive for example with **tar** e.g. into the directory `/opt`:

```
linux:/opt# tar xf /tmp/otrs-2.0.0.tar.gz
linux:/opt# ls
otrs
linux:/opt#
```

Because the modules of OTRS should not be executed with root rights, a new user for OTRS will be added in the next step. The home directory of this new user should be `/opt/otrs`. If your webserver is not running with the same user rights like the new otrs users (this is the case on most systems), you have to add the new otrs user to the group of the web server user:

```
linux:/opt# useradd -d /opt/otrs/ -c 'OTRS user' otrs
linux:/opt# usermod -G nogroup otrs
linux:/opt#
```

Now some demo config files of OTRS have to be copied. The system will later use the copied files. The files are located in `/opt/otrs/Kernel` and `/opt/otrs/Kernel/Config` and have the ending `.dist`.

```
linux:/opt# cd otrs/Kernel/
linux:/opt/otrs/Kernel# cp Config.pm.dist Config.pm
linux:/opt/otrs/Kernel# cd Config
linux:/opt/otrs/Kernel/Config# for foo in *.dist; do cp $foo `basename $foo .dist`; done
linux:/opt/otrs/Kernel/Config#
```

The last step to prepare the installation of OTRS is to set the proper access rights for the files of the ticket system. You can use the script **SetPermissions.sh**, which is located in the `bin` directory in the home directory of the OTRS user. The script can be executed with the following parameters:


```
SetPermissions.sh { Home directory of the OTRS user } { OTRS user } { Web server user }
[ Group of the OTRS user ] [ Group of the web server user ]
```

Is your web server running with the same user rights as OTRS, the command to set the proper access rights is **SetPermissions.sh /opt/otrs otrs otrs**. On SUSE systems the web server is running with the user rights of `wwwrun`. You would use the command **SetPermissions.sh /opt/otrs otrs wwwrun** to set the proper access rights.

3.2.2. Installation of needed perl modules

OTRS needs some additional perl modules. If you install OTRS from source, you'll have to install these modules manually. This can either be done with the packagemanager of your distro (`yast`, `apt-get`) or, like described in this section, through the perl shell and CPAN.

OTRS requires the following additional perl modules:

Table 3-1. Needed perl modules for OTRS

Name	Description
CGI	This module is needed by the web interface of OTRS.
Date::Pcalc	This module is needed for date calculations. OTRS uses this module for example in time specific calculations for tickets.
DBI	OTRS needs this module to connect to the database backend.
DBD::mysql	Module with special functions to connect to the MySQL database backend.
Digest::MD5	This module makes it possible to use the md5 algorithm..
LWP::UserAgent	Module to process HTTP requests.
MIME::Base64	En- and decoding Base64 strings, e.g. for mail attachments.
MIME::Tools	This module provide some tools to process messages with MIME parts.
Mail::Internet	This module makes it possible to process emails based on the RFC 822 standard.
Net::DNS	Perl interface to the domain name system.

Name	Description
Net::POP3	This module contains procedures to access and process messages on a POP3 server.
Net::LDAP	Perl interface to a LDAP directory. You only need to install this module, if you want to use a LDAP back-end.
Net::SMTP	Module that contains procedures to send emails.
Authen::SASL	SASL authentication framework, e.g. needed for the authentication against mailservers.
GD	Interface to the GD graphics library. You only need to install this module, if you want to use the stats module in OTRS.
GD::Text, GD::Graph, GD::Graph::lines, GD::Text::Align	Some more text- and graphic tools for the GD graphics library. You only need to install this modules, if you want to use the stats module of OTRS.
XML::Parser	This module is needed to read and write xml configuration files. The graphical configuration front-end of OTRS uses this module.

To install one of the modules from above with CPAN, you have to execute the command **perl -e shell -MCPAN**. The perl shell will be started in interactive mode and the CPAN module will be loaded. If CPAN is already configured, you can install the modules with the command **install** followed by the name of the module. CPAN takes care of the dependencies of a module to other perl modules and lets you know, if other modules are needed.

After you have installed all modules you can use the script **otrss.checkModules** to check if OTRS has all needed perl modules. The script is located in the `bin` directory in the home directory of the OTRS user.

```
linux:~# cd /opt/otrs/bin/
linux:/opt/otrs/bin# ./otrs.checkModules
      CGI ... ok
    Date::Pcalc ... ok
      DBI ... ok
    DBD::mysql ... ok
    Digest::MD5 ... ok
LWP::UserAgent ... ok
    IO::Scalar ... ok
      IO::Wrap ... ok
    MIME::Base64 ... ok
    MIME::Tools ... ok
Mail::Internet ... ok
      Net::DNS ... ok
```

```

Net::POP3 ... ok
Net::LDAP ... ok
Net::SMTP ... ok
Authen::SASL ... ok
GD ... ok
GD::Text ... ok
GD::Graph ... ok
GD::Graph::lines ... ok
GD::Text::Align ... ok
XML::Parser ... ok
linux:/opt/otrs/bin#

```

Execute also the two commands **perl -cw bin/cgi-bin/index.pl** and **perl -cw bin/PostMaster.pl** after changing into the directory `/opt/otrs`. If the output of both commands is "syntax OK", you have a proper perl installation to use with OTRS.

```

linux:~# cd /opt/otrs
linux:/opt/otrs# perl -cw cgi-bin/installer.pl
cgi-bin/installer.pl syntax OK
linux:/opt/otrs# perl -cw PostMaster.pl
PostMaster.pl syntax OK
linux:/opt/otrs#

```

3.2.3. Configuring the apache web server

This section describes the basic configuration of the apache web server for OTRS. The web server should be able to execute CGI scripts. OTRS won't work if no perl scripts can be parsed. Check the config files of your web server and search for the line that loads the CGI module. If you see something like the following, the CGI module should be in use.

```
LoadModule cgi_module /usr/lib/apache2/modules/mod_cgi.so
```

To access the web interface of OTRS comfortably via a short address, an Alias and a ScriptAlias entry is needed. Most apache installations have a `conf.d` directory included. On linux systems you can find this directory very often under `/etc/apache` or `/etc/apache2`. Log in as root and change to the `conf.d` directory and open/create a file called `otrs.conf`. Insert the following lines into this file:

```
#
```

```
# Basic apache configuration file for OTRS
#
# agent, admin and customer frontend
#
ScriptAlias /otrs/ "/opt/otrs/bin/cgi-bin/"
Alias /otrs-web/ "/opt/otrs/var/httpd/htdocs/"
#
# Directory settings
#
<Directory "/opt/otrs/bin/cgi-bin/">
    AllowOverride None
    Options +ExecCGI -Includes
    Order allow,deny
    Allow from all
</Directory>
<Directory "/opt/otrs/var/httpd/htdocs/">
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

Save the file and restart your webserver to load the new config settings for the web server. On most systems you can start/restart your webserver with the command **/etc/init.d/apache restart** or **/etc/init.d/apache2 restart**.

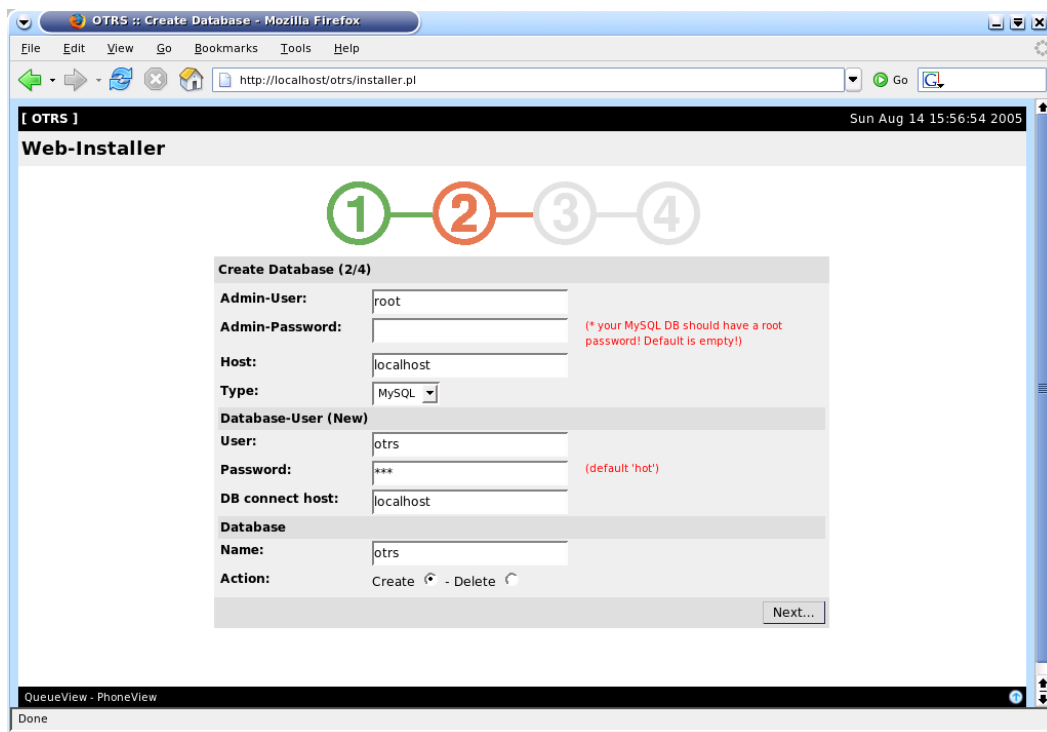
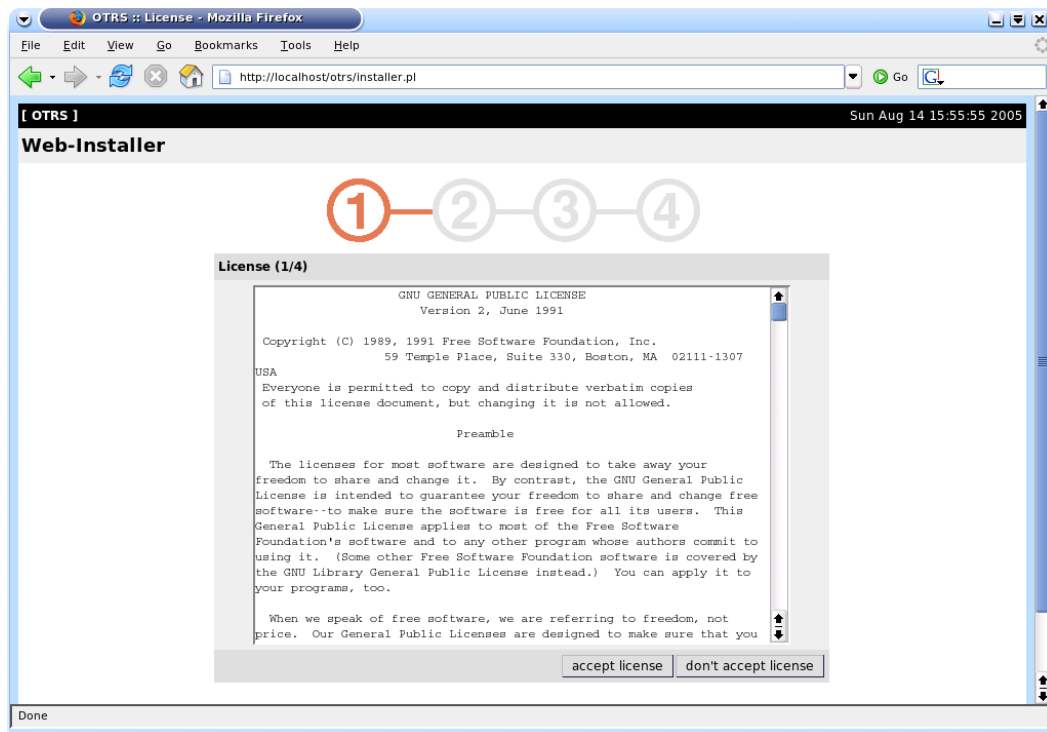
```
linux:/etc/apache2/conf.d# /etc/init.d/apache2 restart
Forcing reload of web server: Apache2.
linux:/etc/apache2/conf.d#
```

Now your webserver should be basicaly configured for OTRS.

3.2.4. Configuring the database

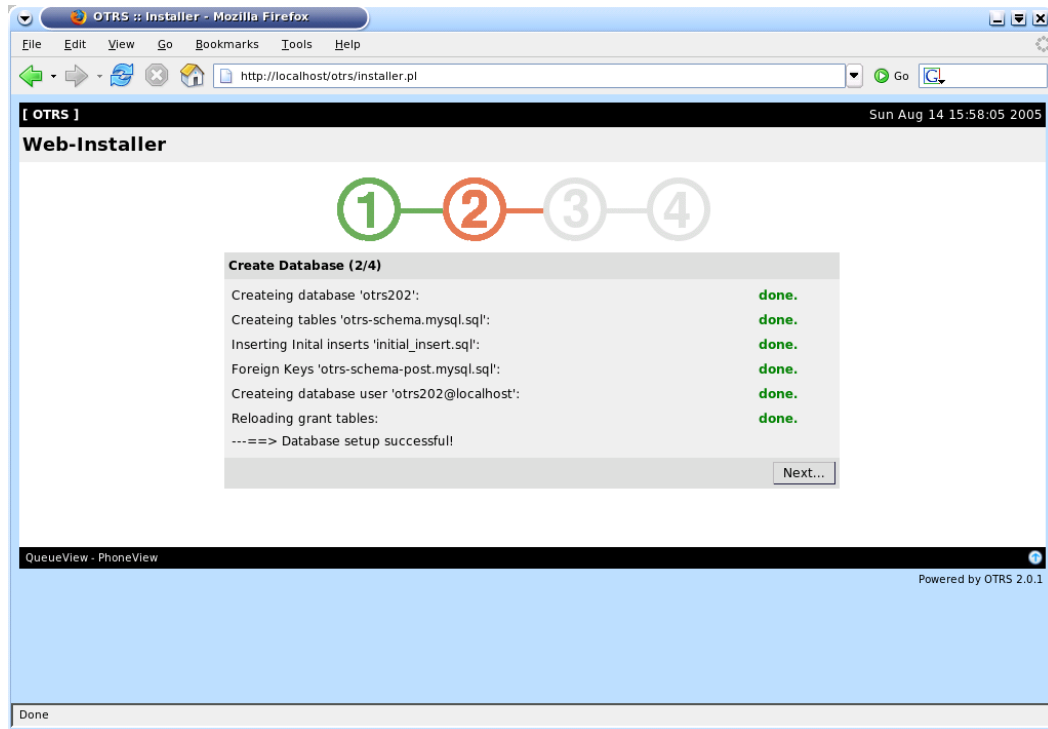
3.2.4.1. The simple way - Using the web installer (works only with MySQL)

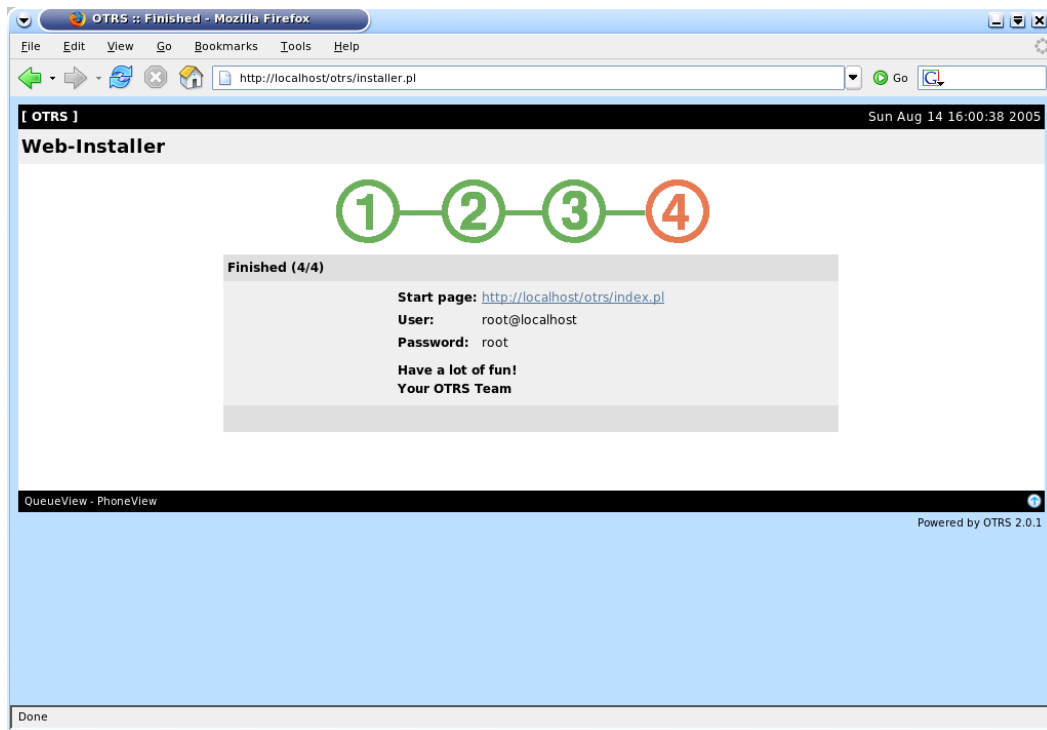
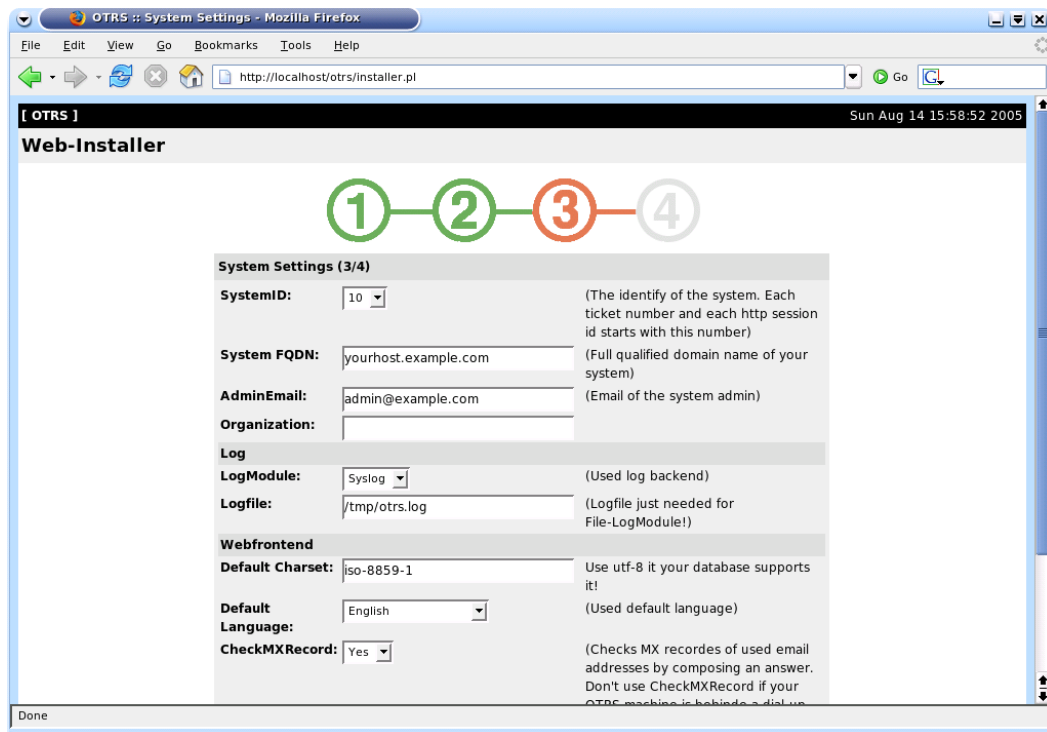
If you are using MySQL for database back-end, you can easily configure the OTRS database via a web front-end. Use the URL <http://localhost/otrs/installer.pl> (<http://localhost/otrs/installer.pl>) to access the start page of the web installer. Just follow the few steps through the installation process.



Warning

It is never a good idea to use default passwords. Please change the default password for the OTRS database!





3.2.4.2. Installing the OTRS database manually

If you can't use the web installer to setup the OTRS database, you have to set it up manually. Scripts with the SQL statements to create and configure the database are located in `scripts/database` in the home directory of the OTRS user.

```
linux:~# cd /opt/otrs/scripts/database/
linux:/opt/otrs/scripts/database# ls
initial_insert.sapdb.sql      otrs-schema-post.maxdb.sql
initial_insert.sql           otrs-schema-post.mysql.sql
otrs-schema.maxdb.sql        otrs-schema-post.oracle.sql
otrs-schema.mysql.sql        otrs-schema-post.postgresql.sql
otrs-schema.oracle.sql       otrs-schema.xml
otrs-schema.postgresql.sql
linux:/opt/otrs/scripts/database#
```

To setup the database for the different database back-ends the `.sql` files must be processed in a special order.

Create the OTRS database manually step by step

1. Creating the DB: Create the database, that you want to use for OTRS, with your database client or your database interface.
2. Creating the tables: With the `otrs-schema.DatabaseType.sql` files (e.g. `otrs-schema.oracle.sql`, `otrs-schema.postgresql.sql`) you can create the tables in your OTRS database.
3. Inserting the initial system data: OTRS needs some initial system data to work properly (e.g. the different ticket states, ticket and notification types). Depending on the type of your database Use either the file `initial_insert.sql` or `initial_insert.sapdb.sql` to insert the initial data.
4. Creating references between tables: The last step is to create the references between the different tables in the OTRS database. Use the `otrs-schema-post.DatabaseType.sql` files to create these references (e.g. `otrs-schema-oracle.post.sql`, `otrs-schema-post.postgresql.sql`).

After you have finished the database setup you should check and set proper access rights for the OTRS database. It should be enough to grant access to one user. Depending on the database server you are using setting up the access rights differs, but it should be possible either with your database client or your graphical database front-end.

If your database and the access rights are configured properly, you have to tell OTRS which database back-end you want to use and how the ticket system can connect to the database. Open the file `Kernel/Config.pm` in the home directory of the OTRS user and change the following parameters to your needs:

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

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

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

3.2.5. Setting up the cron jobs for OTRS

OTRS needs some cron jobs to work properly. The cron jobs should be run with the same user rights that were specified for the OTRS modules. That means that the cron jobs must be inserted into the crontab file of the OTRS user.

All scripts with the cron jobs are located in `var/cron` in the home directory of the OTRS user.

```
linux:~# cd /opt/otrs/var/cron
linux:/opt/otrs/var/cron# ls
aaa_base.dist          pending_jobs.dist      session.dist
fetchmail.dist         postmaster.dist        unlock.dist
generic_agent-database.dist postmaster_pop3.dist
generic_agent.dist     rebuild_ticket_index.dist
linux:/opt/otrs/var/cron#
```

All scripts are ending in `.dist`. You should copy them to files with no ending.

```
linux:/opt/otrs/var/cron# for foo in `ls -l *.dist` ; do cp $foo
`basename $foo .dist`; done
```

```

linux:/opt/otrs/var/cron# ls
aaa_base                generic_agent.dist      rebuild_ticket_index
aaa_base.dist           pending_jobs
rebuild_ticket_index.dist
fetchmail               pending_jobs.dist      session
fetchmail.dist          postmaster              session.dist
generic_agent            postmaster.dist         unlock
generic_agent-database  postmaster_pop3         unlock.dist
generic_agent-database.dist postmaster_pop3.dist
linux:/opt/otrs/var/cron#

```

The following table describes what the several scripts are doing and why they are need to be a cron job for the OTRS user.

Table 3-2. Description of the several cron job scripts

Script	Function
aaa_base	This script sets the basics for the crontab of the OTRS user.
fetchmail	If new mails shall be fetched with fetchmail into the ticket system, this script can be used.
generic_agent	This script executes the jobs of the GenericAgent that are not stored in the database but in own config files.
generic_agent-database	This script executes the jobs of the GenericAgent that are stored in the database.
pending_jobs	This script checks the system for waiting (pending) tickets.
postmaster	This script checks the message queue of the ticket system and delivers messages that are still in the queues.
postmaster_pop3	This script fetches the mails from the POP3 accounts that were specified in the admin area in the section for "PostMaster POP3 Account".
rebuild_ticket_index	This script rebuilds the ticket index. That improves the speed of the QueueView.
session	This script removes old and not longer needed session ID's.
unlock	This script unlocks tickets in the system.

To setup all cron jobs the script `bin/Cron.sh` can be used, which is located in the home directory of the

OTRS user. the script needs a parameter when it is executed that tells if you like to install, remove or reinstall the cron jobs. The following parameters can be used:

```
Cron.sh { start } { stop } { restart } [ OTRS user ]
```

Because the cron jobs need to be installed in the crontab file of the OTRS user, you need to be logged in as OTRS user. If you are logged in as root, you can change to the OTRS user with the command **su otrs**. Execute the following command to install the cronjobs:

Warning

Please note that other crontab entries of the OTRS user will be overwritten or removed by the `Cron.sh` script. Please change the `Cron.sh` script to keep also other crontab entries.

```
linux:/opt/otrs/var/cron# cd /opt/otrs/bin/
linux:/opt/otrs/bin# su otrs
linux:~/bin$ ./Cron.sh start
Cron.sh - start/stop OTRS cronjobs - <$Revision: 1.9 $>
Copyright (c) 2002 Martin Edenhofer <martin@otrs.org>
(using /opt/otrs) done
linux:~/bin$ exit
exit
linux:/opt/otrs/bin#
```

the command **crontab -l -u otrs**, which can be executed as root, shows you the crontab file of the OTRS user and you can check if all entries are right:

```
linux:/opt/otrs/bin# crontab -l -u otrs
# --
# cron/aaa_base - base crontab package
# Copyright (C) 2002-2003 Martin Edenhofer <martin+code@otrs.org>
# --
# $Id: installation-and-basic-configuration.xml,v 1.9 2006/08/21 13:34:29 cs Exp $
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# Who gets the cron emails?
MAILTO="root@localhost"

# --
```

```
# cron/fetchmail - fetchmail cron of the OTRS
# Copyright (C) 2002-2003 Martin Edenhofer <martin+code@otrs.org>
# --
# $Id: installation-and-basic-configuration.xml,v 1.9 2006/08/21 13:34:29 cs Exp $
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# fetch every 5 minutes emails via fetchmail
*/5 * * * * /usr/bin/fetchmail -a >> /dev/null

# --
# cron/generic_agent - GenericAgent.pl cron of the OTRS
# Copyright (C) 2002-2003 Martin Edenhofer <martin+code@otrs.org>
# --
# $Id: installation-and-basic-configuration.xml,v 1.9 2006/08/21 13:34:29 cs Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# start generic agent every 20 minutes
*/20 * * * * $HOME/bin/GenericAgent.pl >> /dev/null
# example to execute GenericAgent.pl on 23:00 with
# Kernel::Config::GenericAgentMove job file
#0 23 * * * $HOME/bin/GenericAgent.pl -c "Kernel::Config::GenericAgentMove" >> /dev/null
# --
# cron/generic_agent - GenericAgent.pl cron of the OTRS
# Copyright (C) 2001-2004 Martin Edenhofer <martin+code@otrs.org>
# --
# $Id: installation-and-basic-configuration.xml,v 1.9 2006/08/21 13:34:29 cs Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# start generic agent every 10 minutes
*/10 * * * * $HOME/bin/GenericAgent.pl -c db >> /dev/null
# --
# cron/pending_jobs - pending_jobs cron of the OTRS
# Copyright (C) 2002-2003 Martin Edenhofer <martin+code@otrs.org>
# --
# $Id: installation-and-basic-configuration.xml,v 1.9 2006/08/21 13:34:29 cs Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# check every 120 min the pending jobs
45 */2 * * * $HOME/bin/PendingJobs.pl >> /dev/null
# --
# cron/postmaster - postmaster cron of the OTRS
# Copyright (C) 2002-2003 Martin Edenhofer <martin+code@otrs.org>
# --
# $Id: installation-and-basic-configuration.xml,v 1.9 2006/08/21 13:34:29 cs Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# check daily the spool directory of OTRS
#10 0 * * * * test -e /etc/init.d/otrs & /etc/init.d/otrs cleanup >> /dev/null; test -e /et
10 0 * * * $HOME/bin/otrs.cleanup >> /dev/null
```

```
# --
# cron/postmaster_pop3 - postmaster_pop3 cron of the OTRS
# Copyright (C) 2002-2003 Martin Edenhofer <martin+code@otrs.org>
# --
# $Id: installation-and-basic-configuration.xml,v 1.9 2006/08/21 13:34:29 cs Exp $
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# fetch emails every 10 minutes
*/10 * * * * $HOME/bin/PostMasterPOP3.pl >> /dev/null
# --
# cron/rebuild_ticket_index - rebuild ticket index for OTRS
# Copyright (C) 2002-2003 Martin Edenhofer <martin+code@otrs.org>
# --
# $Id: installation-and-basic-configuration.xml,v 1.9 2006/08/21 13:34:29 cs Exp $
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# just every day
01 01 * * * $HOME/bin/RebuildTicketIndex.pl >> /dev/null

# --
# cron/session - delete old session ids of the OTRS
# Copyright (C) 2001-2004 Martin Edenhofer <martin+code@otrs.org>
# --
# $Id: installation-and-basic-configuration.xml,v 1.9 2006/08/21 13:34:29 cs Exp $
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# delete every 120 minutes old/idle session ids
55 */2 * * * $HOME/bin/DeleteSessionIDs.pl --expired >> /dev/null

# --
# cron/unlock - unlock old locked ticket of the OTRS
# Copyright (C) 2002-2003 Martin Edenhofer <martin+code@otrs.org>
# --
# $Id: installation-and-basic-configuration.xml,v 1.9 2006/08/21 13:34:29 cs Exp $
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# unlock every hour old locked tickets
35 * * * * $HOME/bin/UnlockTickets.pl --timeout >> /dev/null

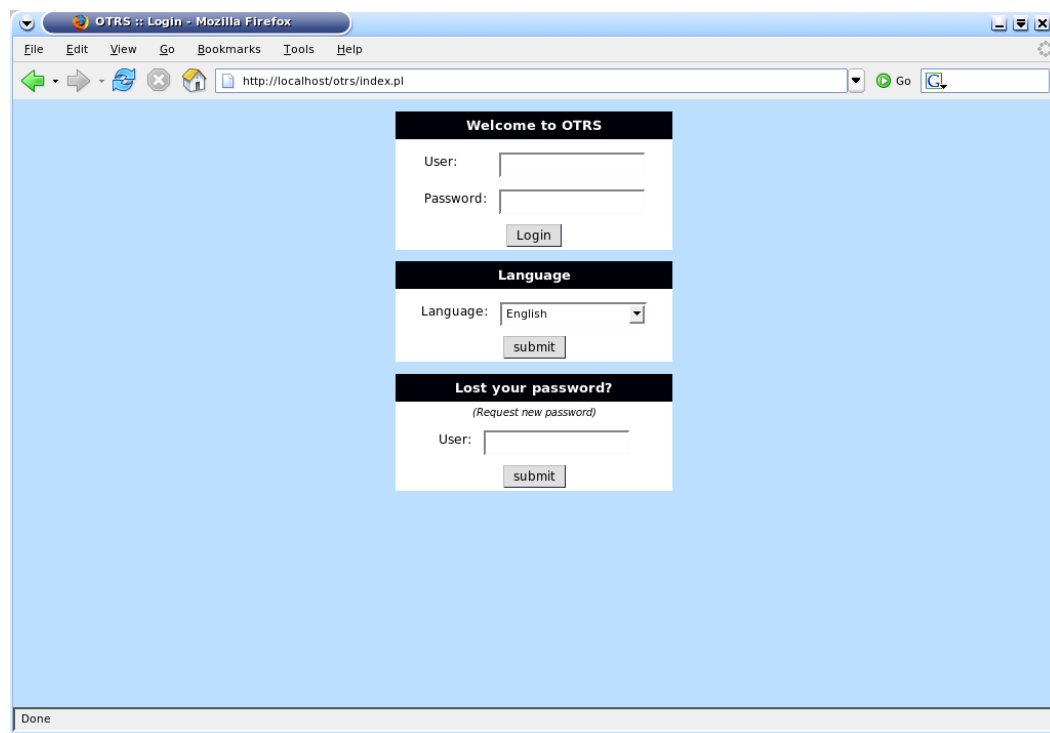
linux:/opt/otrs/bin#
```

Chapter 4. First steps in OTRS

This chapter's goal is to give a quick overview about how OTRS is working and how the web interface is structured. The differences between agents, customers and administrators are explained, and the first login as OTRS admin will be accomplished to take a closer look at the user preferences which are available for every account.

4.1. Agent web interface

All agents use the agent web interface to work with OTRS. Agents answer the customer requests, create new tickets for customers or other agents, write tickets about telephone calls with customers, write FAQ entries or edit customer data, e.g.

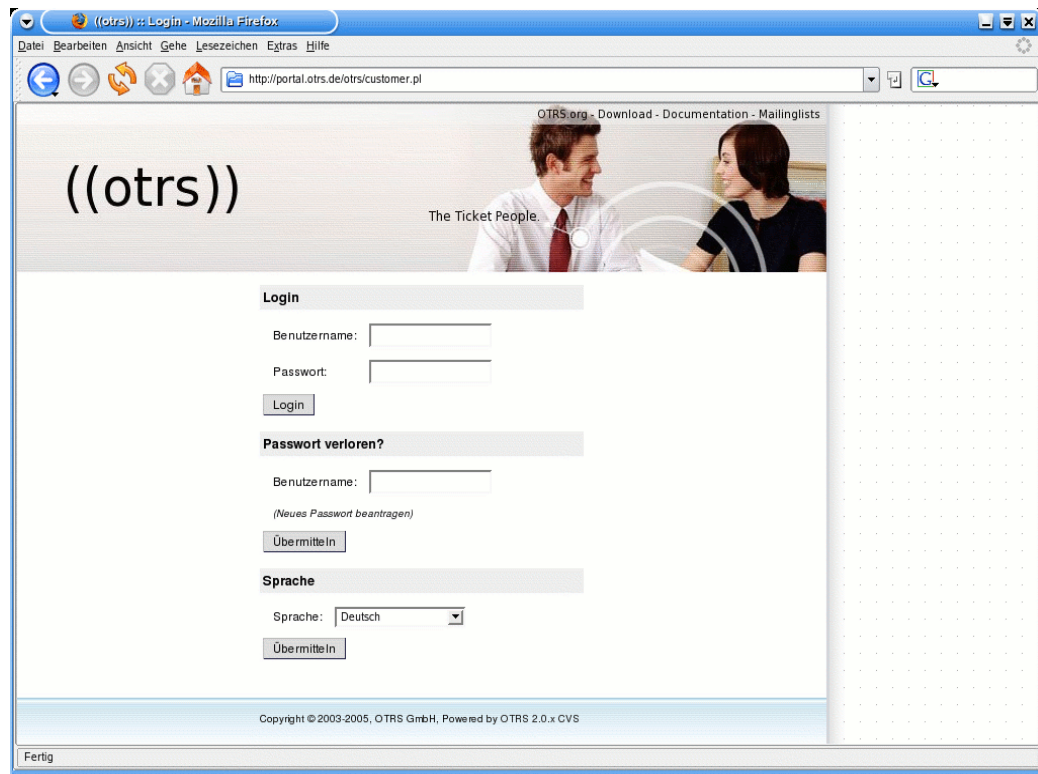


Lets say that your OTRS host is reachable via the URL *http://www.example.com* (*http://www.example.com/*) then the agents and also the OTRS administrator can reach the login screen

by using the address <http://www.example.com/otrs/index.pl> (<http://www.example.com/otrs/index.pl>) in a web browser.

4.2. Customer web interface

Customers have a special web interface in OTRS. Through this web interface customers can create new accounts, get an overview on own tickets, create and edit tickets, change the account settings, e.g.

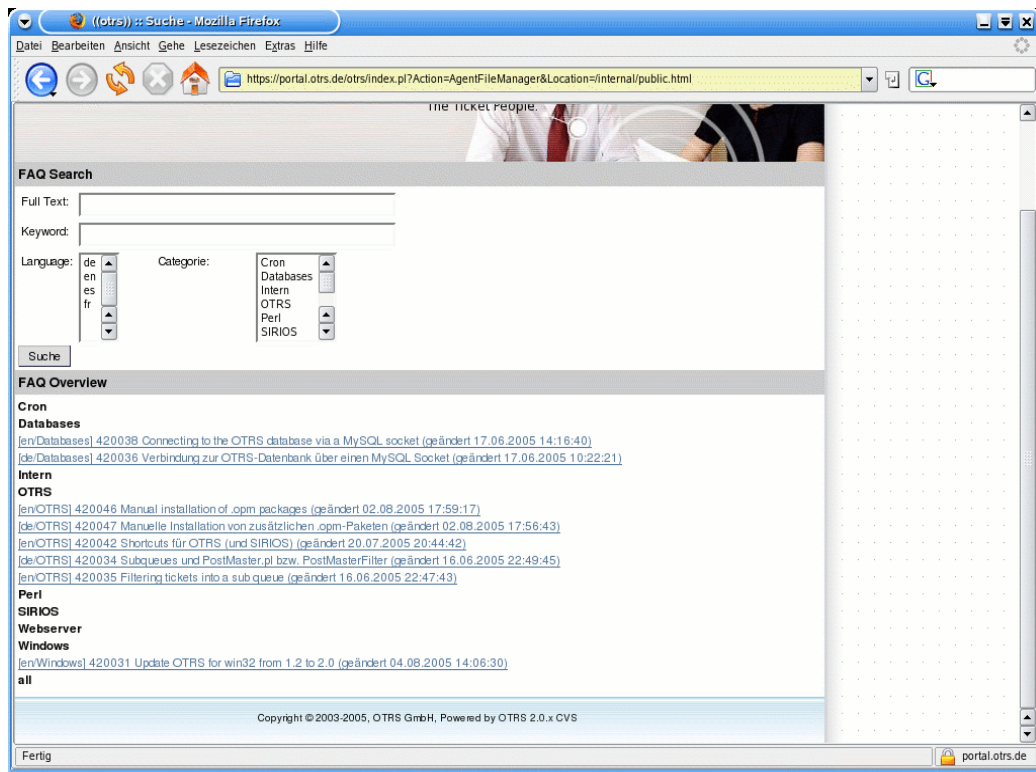


Lets say the host with the OTRS installation is reachable via the domain <http://www.example.com> (<http://www.example.com/>) then the login screen for customers can be reached by the URL <http://www.example.com/otrs/customer.pl> (<http://www.example.com/otrs/customer.pl>) with a web browser.

4.3. Public web interface

In addition to the web interfaces for agents and customers OTRS has a public web interface. It provides

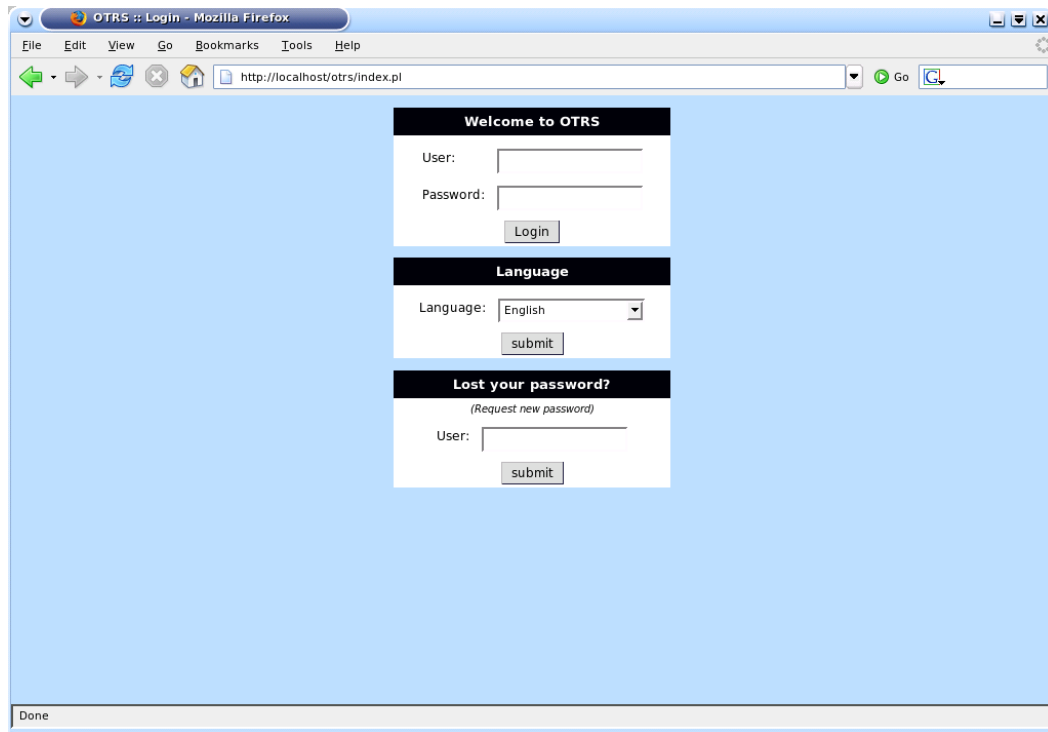
public access to the FAQ system, and lets visitors search through the FAQ entries without any authorization.



The public web interface can be reached via the URL <http://www.example.com/otrs/faq.pl> (<http://www.example.com/otrs/faq.pl>). Also the address <http://www.example.com/otrs/public.pl> (<http://www.example.com/otrs/public.pl>) makes it possible to access the FAQ system.

4.4. First login

Like described in the section on the agent web interface the login screen for agents and the OTRS administrator can be reached through the address <http://www.example.com/otrs/index.pl> (<http://www.example.com/otrs/index.pl>).



The screen lets you enter a user name and a password. Because no users are created after a fresh installation of the system, you have to login as OTRS administrator first. To login as OTRS admin use "root@localhost" for user name and "root" for password.

Warning

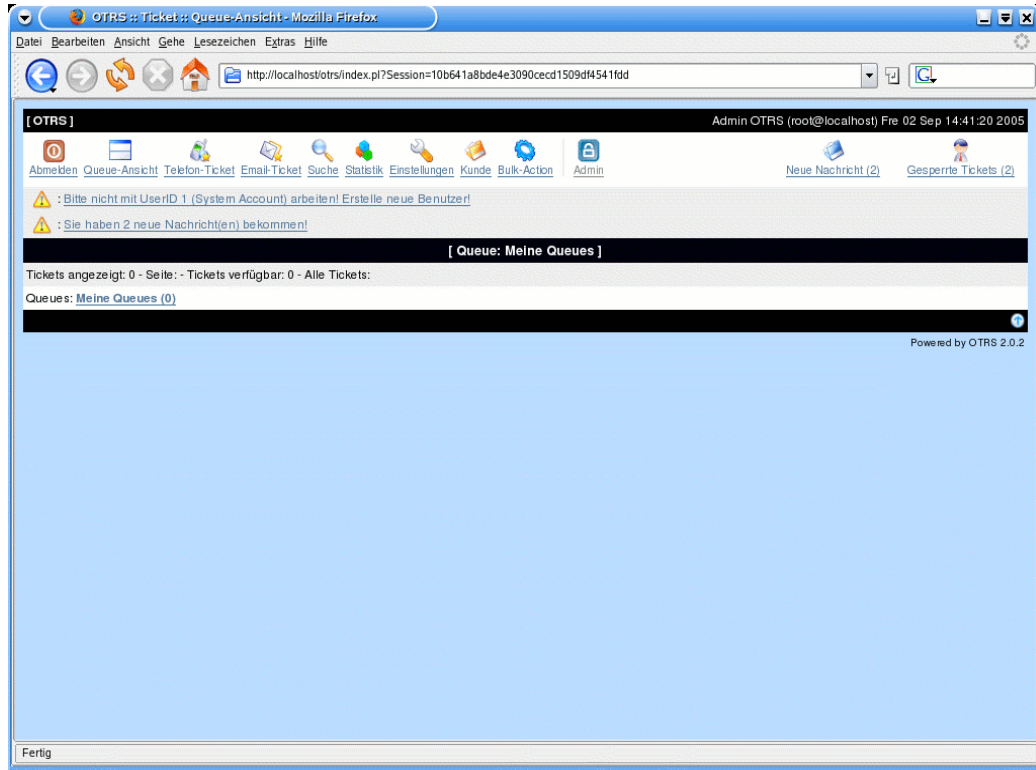
This account data are default on every new installed OTRS system. For that reason you should change the password for the OTRS administrator as fast as possible! This can be done via the preferences for the OTRS administrator account.

If you don't want to login as OTRS administrator just enter the user name and password for your normal agent account. With the list box below the input fields for the user name and password you can select the language that you want to use in the web interface.

If you have forgotten your password, you can get a new password through the system. Just enter the mail address that is registered for you OTRS account into the input field at the lower part of the login screen and submit the input.

4.5. An overview to the web interface

After you have successfully logged into the system the web interface is loaded. Per default you are in the QueueView after the login. The QueueView provides you a quick overview about tickets in the different queues, notifies about new messages, tells you how many locked tickets you have, e.g.



To improve the clarity the web interface is separated into different areas. The black bar on top of the windows shows some general information like the current time and date, your name and your mail address. Also on the left side a link is displayed that can be used to reload the page.

The white bar below is the navigation bar. It shows buttons that enables you to navigate to different areas or modules of the system and lets you execute some global actions. The navbar is divided into three parts. In the left area the logout button, the button to activate the QueueView, a button to load the customer back-end and a button to load the full-text search are displayed. The "Phone-Ticket" button and "Email-Ticket" button lets you create a new phone or email ticket. The stats button is leading to a screen where some system stats can be generated. With the preference button you can reach the screen to change your own account settings. "Bulk-Action" opens a screen to execute some actions on marked tickets, e.g. to close or move more than one ticket.

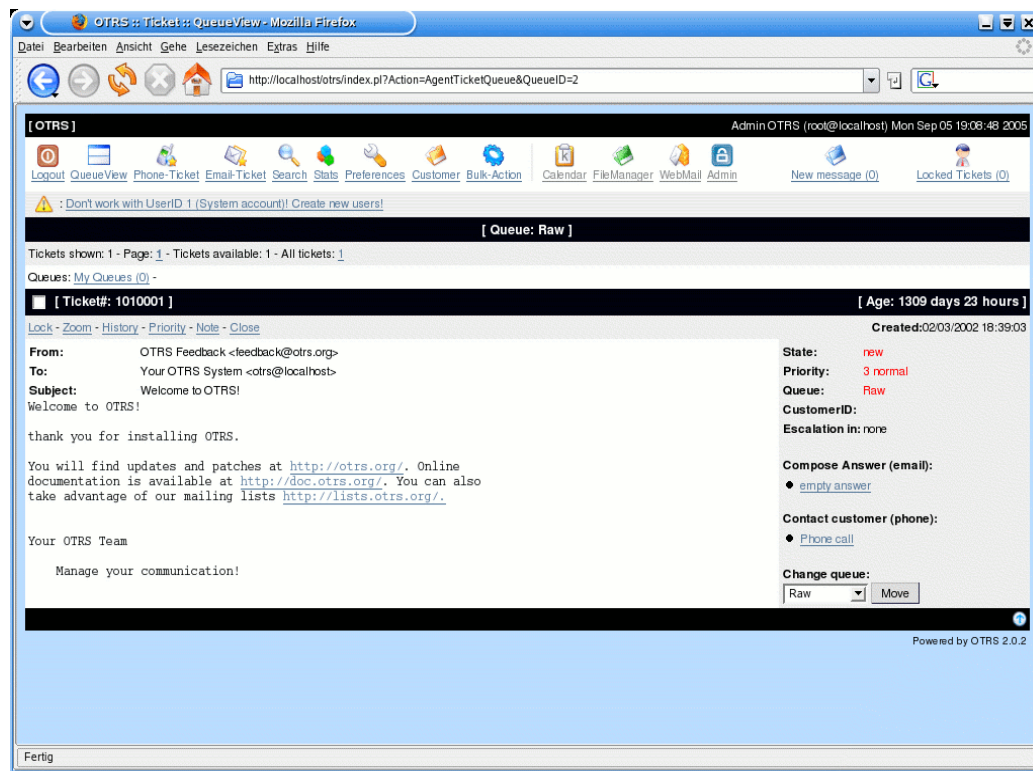
The middle part of the navbar shows the buttons for the navigation to the other modules of the system. After a standard installation you only see the "Admin" button, if you are logged in as OTRS administrator. If some additional application are installed, e.g. the file manager or the web mailer, buttons to reach this applications are also displayed.

At the right site of the navbar you can get an overview on how many tickets you have locked and if new messages for you have been arrived.

The grey bar below the navbar shows different system messages. If you are logged in as OTRS administrator you get a message, that you should not work as administrator. If the calendar module is installed, the next events are displayed in this section.

The black bar below shows the queue, which is currently selected or tells you, that "My Queues" is selected. "My Queues" are the queues, that you want to observe more attentive. You can select the queues for "My Queues" in your account preferences .

Below the area with the currently selected queue a grey bar is displayed that contains informations about the tickets in the system. The next bar shows all queues and their number of open tickets. On a fresh installed system you can see, that the "raw" queue contains one new ticket and "My Queues" is empty.



If you select the "raw" queue, the interface will be reloaded and the content of the queue is displayed. Now the lower part of the screen contains additional information about the new ticket in the "raw" queue. A black bar is filled with the ID of the ticket and its age. At the left side of this bar the checkbox is available that lets you mark the ticket for the bulk action. The next grey line contains the actions that can be done with the ticket (e.g. Lock, Zoom, History). On the right side of this line the date and time when the ticket was created is displayed.

Below the interface is divided into two parts. On the left side a preview of the ticket is displayed in white color. The sender of the ticket, the recipient address, the subject and the first lines are shown. On the right side some additional information for the ticket are displayed in grey color, e.g. the priority and the status. Some buttons allow to answer the ticket, make a telephone note or move the ticket into another queue.

On the bottom of the screen the site footer is displayed in black color. It also contains some buttons to activate the QueueView, jump to the beginning of the screen, e.g.

4.6. What is a queue?

Because queues are very important in OTRS, this section tries to explain more detailed the meaning of queues.

On common mail systems all messages are saved into an inbox file. An inbox is a big file where the messages are arranged on to another. New messages are appended at the end of the inbox file. The mail program, which is used to read and write mails, reads the inbox file and presents the content to the user.

A queue in OTRS is somehow comparable to an inbox file, but has some more features. It also can store many messages, but the mails are saved in another way. You don't need to know any detail of queues if you only want to use OTRS, the only important thing to know is in which queue a ticket is stored. The users of OTRS, better called agents, can open and edit tickets in a queue and of course they can move tickets from one queue to another. But why should they move tickets?

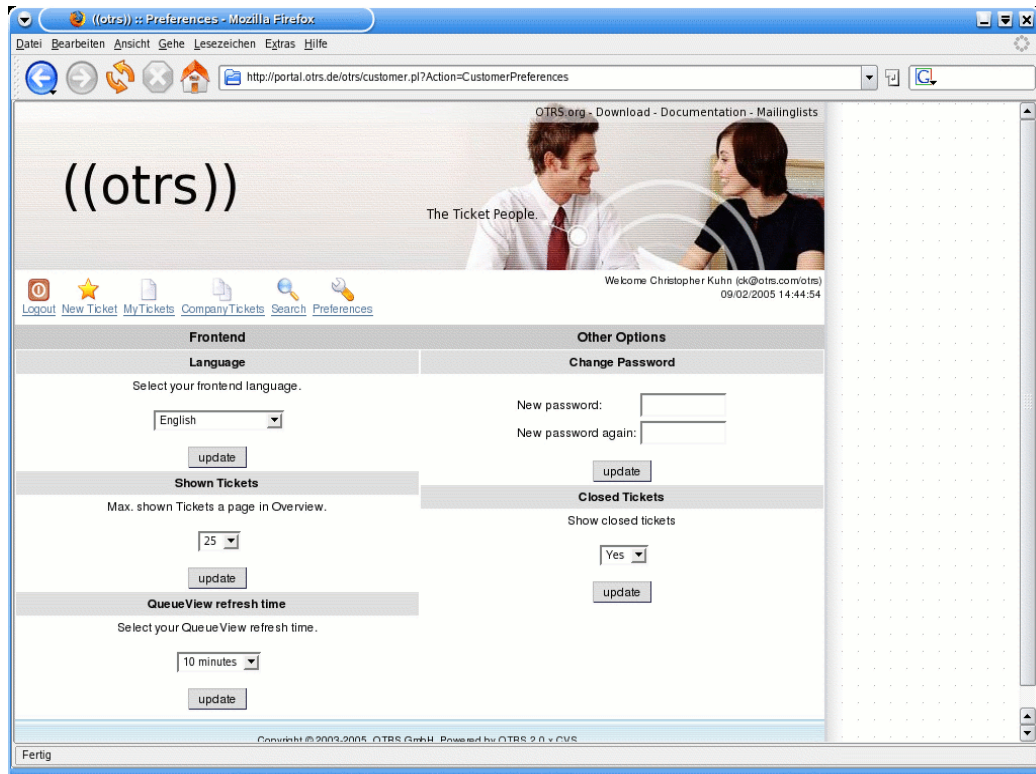
To explain it more practically, remember Max's company describe in the section about a example of a ticket system. Max' has installed OTRS to get rid of his support chaos. He and his agents are using the system to manage the support requests for the video recorders. One queue that contains all request is enough for this situation. But some times later Max also sales a DVD recorder. Now the customer not only have questions about the video recorder but also for the new product. More and more emails get into the one queue of Max's OTRS and its hard to keep the overview. Some day Max decides to optimize the structure of his support system and adds two new queues, so now three queues are used by the system. Into the old queue (called "raw") all mails are stored if they arrive at the ticket system. The two new queues are one for the video recorder (called "video recorder") and the other one for the dvd recorder requests (called "dvd recorder"). Max tells Sandra to watch the "raw" queue and sort (dispatch) the mails

either into "video recorder" or "dvd recorder", depending if a customer asks for support for a video recorder or dvd recorder. John only has access to the "video recorder" queue, Joe can only answer tickets in the "dvd recorder" queue. Max is able to edit tickets in all queues. Because OTRS supports access management for users, groups and roles, it is easy to setup queues that only are accessible for some specific accounts. Max could also use another way to get his requests into the different queues, with filter rules or if two different mail addresses are used Sandra only has to dispatch emails into the two other queues, that can't be dispatched automatically.

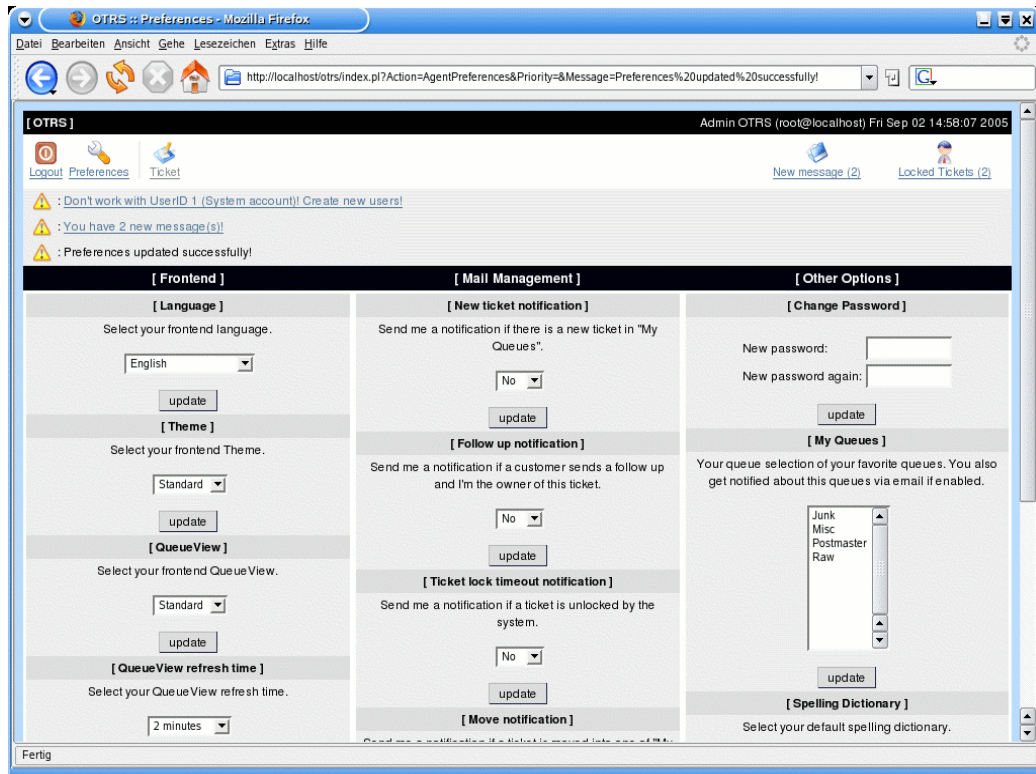
Sorting your incoming messages into different queues helps you to keep the support system structured and tidy. Because your agents are arranged into different groups with different access rights onto queues, the system can be optimized even more. Queues can be used to define work flow processes or they create the structure of a company. Max could implement for example another queue called "sales". this queue could contain the sub queues "requests", "offers", "orders", "billing", e.g. Such a queue structure could help Max to optimize his order transactions. The better a system is structured, the less time is needed for the different tasks. That results in less working time that is needed for the different tasks and that saves money. Queues can help to optimize the processes in your company.

4.7. User preferences

Many settings of a user account in OTRS can be configured to their own needs via the user preferences. Customers, agents and the OTRS administrator have the possibility to adjust their own account preferences. The preferences page can be reached via the "Preferences" link in the customer or agent interface.



A customer can select the web interface language, the maximal count of shown tickets and the refresh period for the interface. Also it is possible to set a new password and activate/deactivate the display of closed tickets.



An agent can adjust the interface language, the theme, the default dictionary and the view for the queues. Also the refresh period for the interface, the maximal count of shown tickets per page and the screen that should be displayed after a new ticket is created can be set. It is possible to change the password and select the events when the system shall send a notification to the email address, that is registered for your account.

The queues, that you want to monitor in your "My Queues" settings, can also be selected in the user preferences. You should only select such queues that are important for you.

Chapter 5. The admin area of OTRS

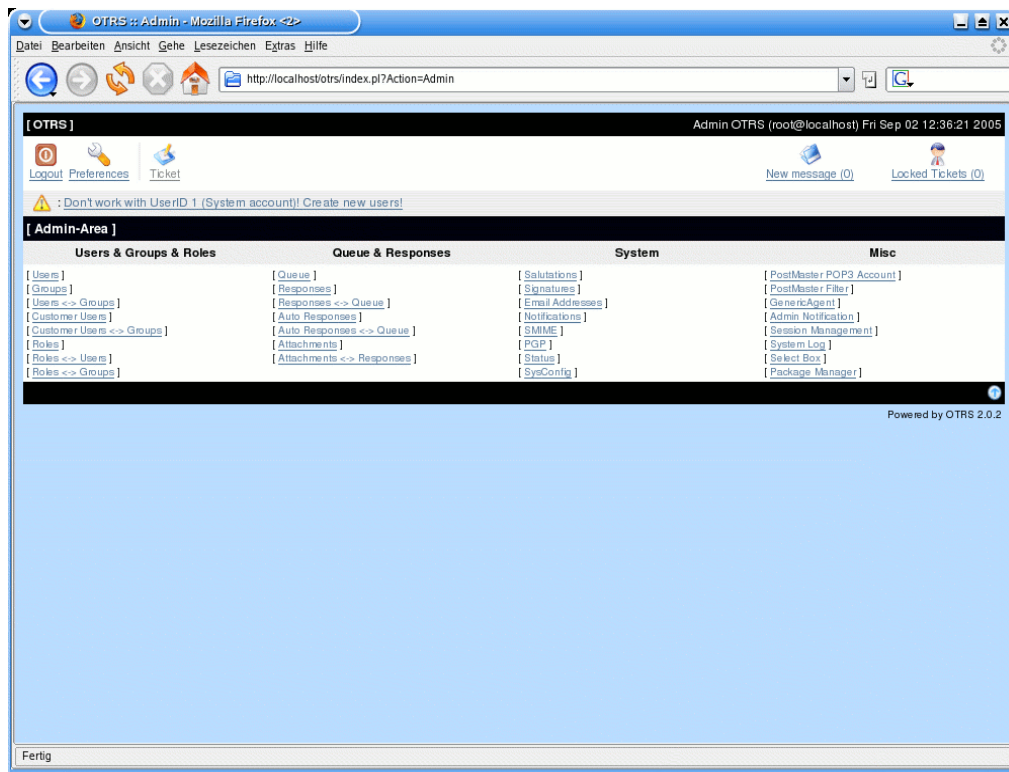
5.1. Basics

The admin area of OTRS is the central part in the web interface for the OTRS administrator. All important settings for the system can be done, changed and viewed via this module.

The admin area is reachable via the "Admin" link in the navbar of the agent interface. The link is only displayed, if you are logged in as OTRS administrator or if you are a member of the admin group. After a default installation you can log in as OTRS admin with the username root@localhost and the password root.

Warning

Please change this password as fast as possible via the user preferences page, because this password is set per default on new OTRS systems.



5.2. User, groups and roles

5.2.1. User

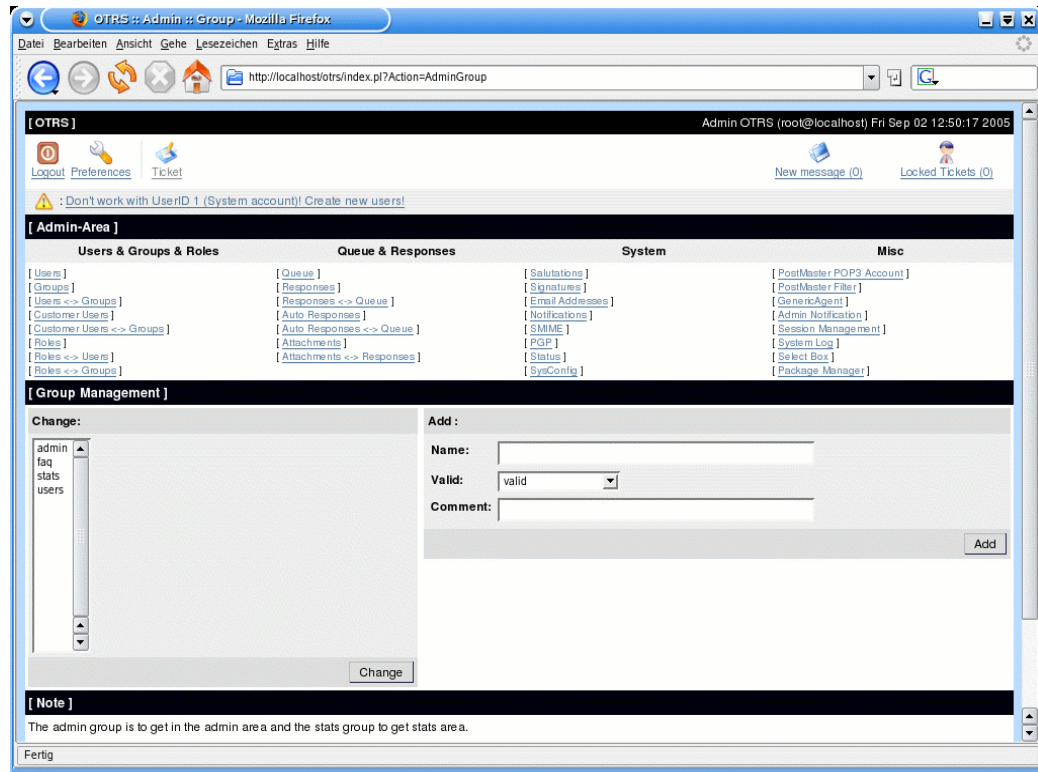
Via the link "User" the user management of OTRS can be reached. It is possible to add, change or deactivate user accounts. Also some basic setting for a user account can be adjusted, e.g. the language for the web interface or the theme.

Note: To keep the consistency of the data for OTRS it is not possible to delete a user. To deactivate a user account set it to "invalid" or "invalid-temporarily".

After a new user has been created the user must be added to one or more groups or roles. You will be redirected automatically to the screen for the group and user management when a new account has been created.

5.2.2. Groups

Every user account should at least belong to one group. The group management can be reached via the "Groups" link.



Note: To keep the consistency of the data for OTRS it is not possible to delete a group. To deactivate a group set it to "invalid" or "invalid-temporarily".

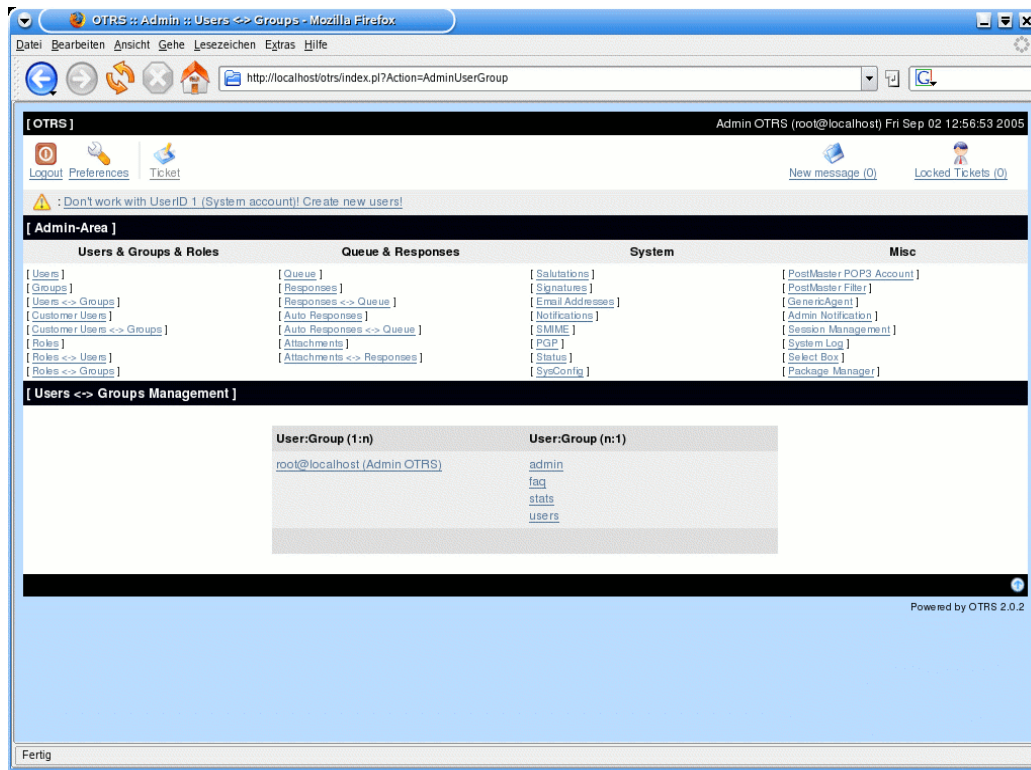
After a default installation already four groups are available in the system.

Table 5-1. Default groups after OTRS has been installed

Group	Description
admin	Group for the users who should have admin rights in the system. After the system has been installed only the user "root@localhost" is in this group.

Group	Description
faq	Users in this group may create and change articles in the FAQ system. After the system has been installed no user is in this group.
stats	Users in this group may access the stats module of OTRS and generate statistics. After the system has been installed only "root@localhost" belongs to this group.
users	This is the group where your agents should belong to and have read and write access. If users are in this group and have write rights they can use all functions of the ticket system. After the system has been installed this group is empty.

To add a user to a group or to change the group settings of a user the link "Users <-> Groups" can be used.



An overview of all groups and users in the system is displayed in the lower part of the screen. If you want to change the group settings of a user just click on the username. To change the users in a group just click on the group that you want to change.

To setup the rights for a user very gradually there are some rights that can be set for a user account. After a default installation the following rights are available.

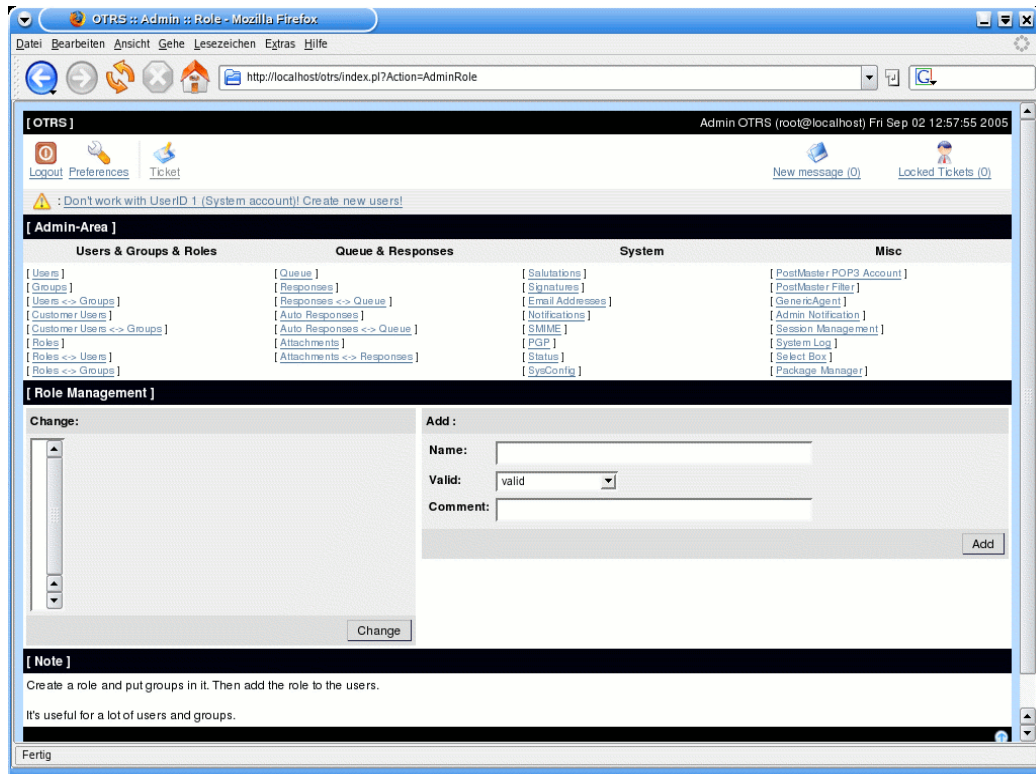
Table 5-2. Rights in the user groups of OTRS

Right	Description
ro	Read only access to the tickets and entries or queues of this group.
move into	Right to move tickets or entries between queues or areas that belong to this group.
create	Right to create tickets or entries in the queues or areas of this group.
owner	Right to update the owner of tickets or entries in queues or areas that belong to this group.
priority	Right to change the priority of a ticket or entry in queues or areas that belong to this groups.
rw	Full read and write access on all tickets or entries in the queues or areas that belong to this group.

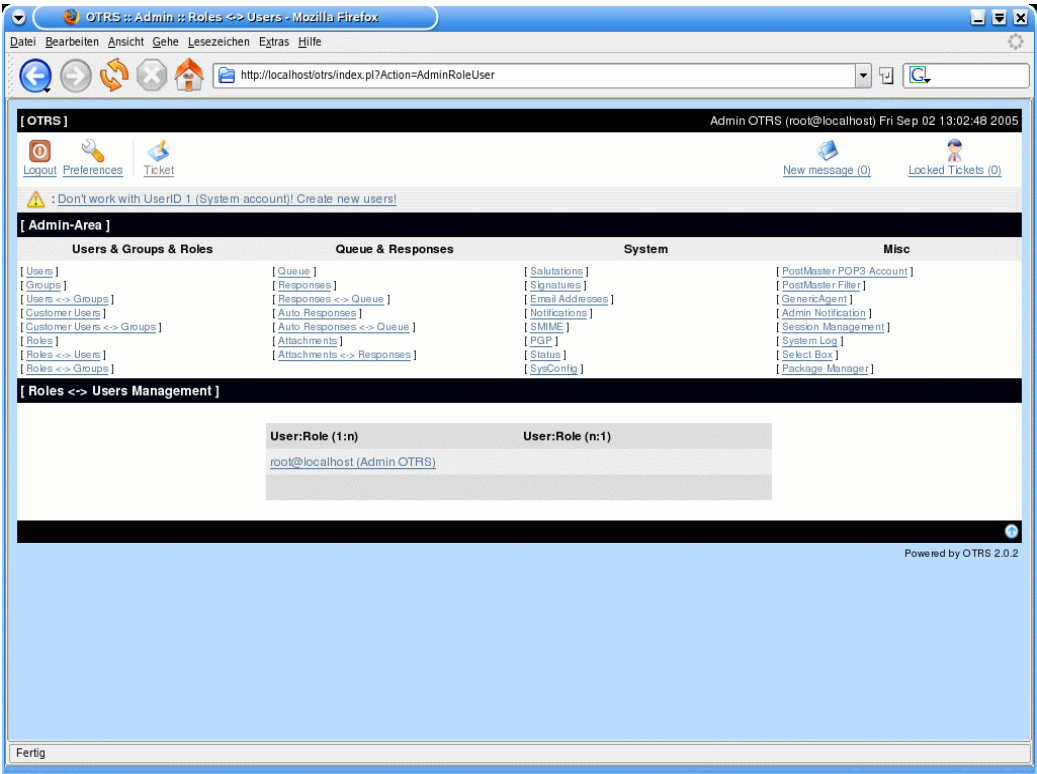
5.2.3. Roles

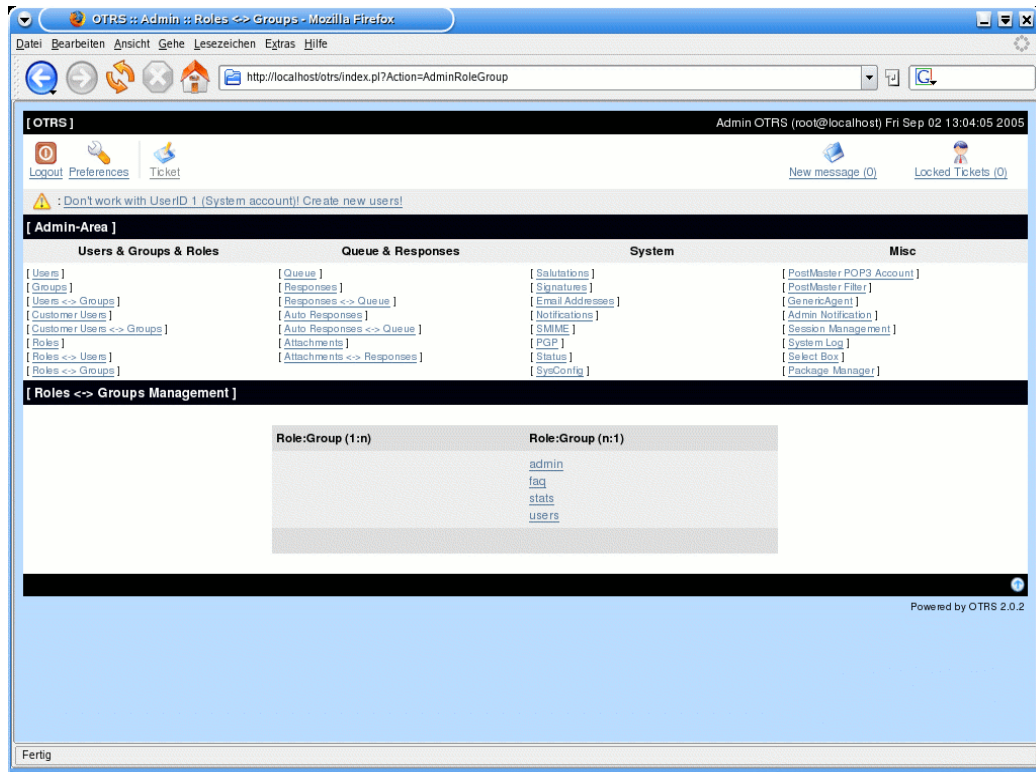
Roles are a very powerful and helpful feature to manage and change the access rights of many users very simply and quickly. On big and complex systems with many users, groups and queues this feature is very useful and helps to save time.

To explain the advantages of the role feature imagine that you have a system with 100 users. 90 users have access to a queue called "support" that contains some sub queues. All support requests are handled through this queue. All other queues of the system are not accessible for the 90 users. The other 10 users may access all queues of the system, they are not restricted only to the support queue. They dispatch tickets, watch the raw queue and move spam messages into the "junk" queue. Some day the company opens a new department that sells products. Offerings, confirmations of offerings, bills, e.g. must be created and some of the agents in your system shall do this via OTRS. The different users have to get access to the new queues that must be created. Because it would take a long time to change the access rights for the different users manually, roles can be created that define the different access levels. Then the users can be added to one or more roles and the user rights get for this user accounts get changed. If a new user account is created it is also possible to add this account to one or more roles.



Note: To keep the consistency of the data for OTRS it is not possible to delete a role. To deactivate a role set it to "invalid" or "invalid-temporarily".





To define the different access rights for a role the links "Roles <-> Users" or "Roles <-> Groups" can be used.

5.3. Customer users and customer groups

5.3.1. Customer users

OTRS supports different types of users. The link "Customer Users" can be used to manage the accounts of your customers. A customer can log in the system via the customer interface (customer.pl). The customer interface enables a user to manage own tickets and change the own account settings. Also a customer user is needed for the ticket history in the system.

You can search for a specific customer user in the database and change the customer user backend. More infos about customer user backends are available in the chapter about external backends .

You can create new customer user accounts. All input fields that are marked through an asterisk (*) have to contain values. Very important is to specify a account name and a password for the new user to enable that this user can log in the system. Also you have to specify a customer ID ("Customer#"). This ID needs the system to identify the user and the tickets of this user. Ffor example the mail address of the user can be specified for Customer#, because a mail address is explicit.

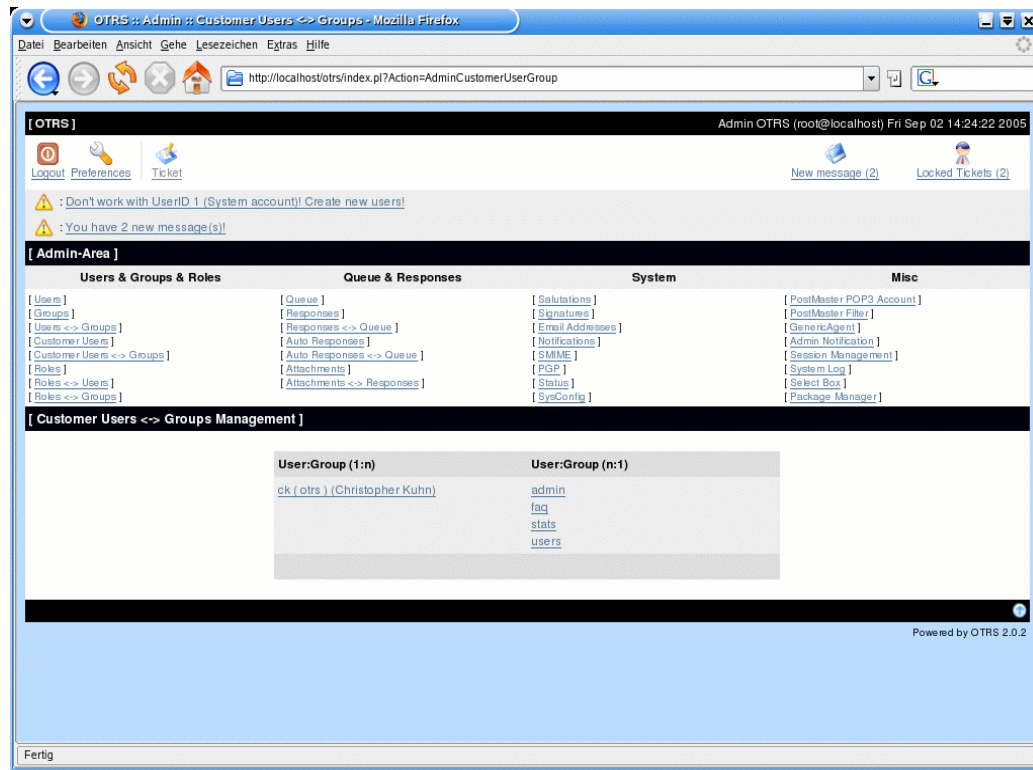
Note: To keep the consistency of the data for OTRS it is not possible to delete a customer user. To deactivate a customer user account set it to "invalid" or "invalid-temporarily".

5.3.2. Customer groups

Customer users can also be added to a group. this feature might be useful, if you want to add some customer users of the same company that shall only have access to one or some queues through the

customer interface. First create via the group management module the group, that shall contain all customer users of the company. Then create the queues and select the new group for this queues.

The next step is to activate the customer group support. This can be done with the configuration parameter `CustomerGroupSupport` either via the graphical configuration frontend of OTRS or via inserting this parameter into your `Kernel/Config.pm` file. The parameter `CustomerGroupAlwaysGroups` you can specify the groups for a new added customer user, the new account will be added to this group automatically.



Through the link "Customer Users <-> Groups" you can manage which customer user shall belong to the different groups.

5.4. Queues

Through the link "Queue" you can manage the queues of your system. In a new installed system the "Junk", "Misc", "Postmaster" and "Raw" queues already exist in your system. "Raw" is the default

queue, all incoming messages are stored in this queue if no filter rules are defined. The "Junk" queue can be used to store spam messages.

[Queue Management]

Change:

- Junk
- Misc
- Postmaster
- Raw

Add:

Name:

Sub-Queue of:

Group:

Unlock timeout (minutes): 0 = no unlock - 1 day = 1440 minutes

Escalation time (minutes): 0 = no escalation - 1 day = 1440 minutes

Follow up Option:

Ticket lock after a follow up:

Systemaddress:

Salutation:

Signature:

Customer Move Notify:

Customer State Notify:

Customer Owner Notify:

Valid:

Comment:

[Note]

Key	Description
Unlock timeout	If an agent locks a ticket and he/she will not send an answer within this time, the ticket will be unlock automatically. So the ticket is viewable for all other agents.
Escalation time	If a ticket will not be answered in thos time, just only this ticket will be shown.
Ticket lock after a follow up	If a ticket is closed and the customer sends a follow up the ticket will be locked for the old owner.

Fertig

The form on the right site of the screen lets you add a new queue. You can specify the name and the group who should use the queue and if you want the new queue to be a sub queue of another already existing queue.

If a ticket was locked by an agent, you can specify a time interval when the ticket is unlocked automatically by the system.

If you specify a escalation time tickets which are older than this time block newer tickets to and the escalated tickets have to be processed first. This setting is usefull to force the processing of older tickets, new tickets don't appear in the QueueView if a ticket is escalated.

Also you can specify that the old owner of a processed ticket is automatically the owner again, if a followup of the old ticket is send to the ticket system. This feature makes sure that a folowup for a ticket is first shown to the origianl owner of the ticket who already knows the facts of this ticket.

The parameter for the system address sets the mail address that is used for the outgoing tickets of this queue. With the salutation and signature parameter the used values for the new queue can be selected. The sections `email addresses` `salutations` and `signatures` explained more detailed this different parameters.

The different customer info parameters let you specify the queue events when a notification to a customer is send.

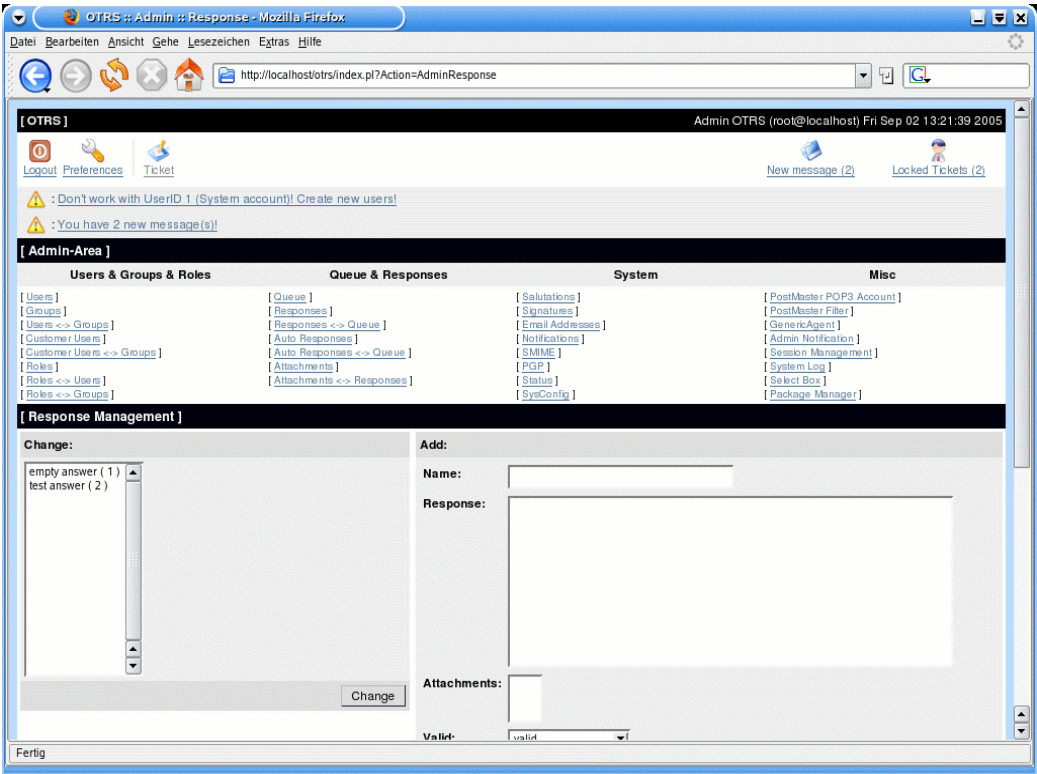
Note: To keep the consistency of the data for OTRS it is not possible to delete a queue. To deactivate a queue set it to "invalid" or "invalid-temporarily".

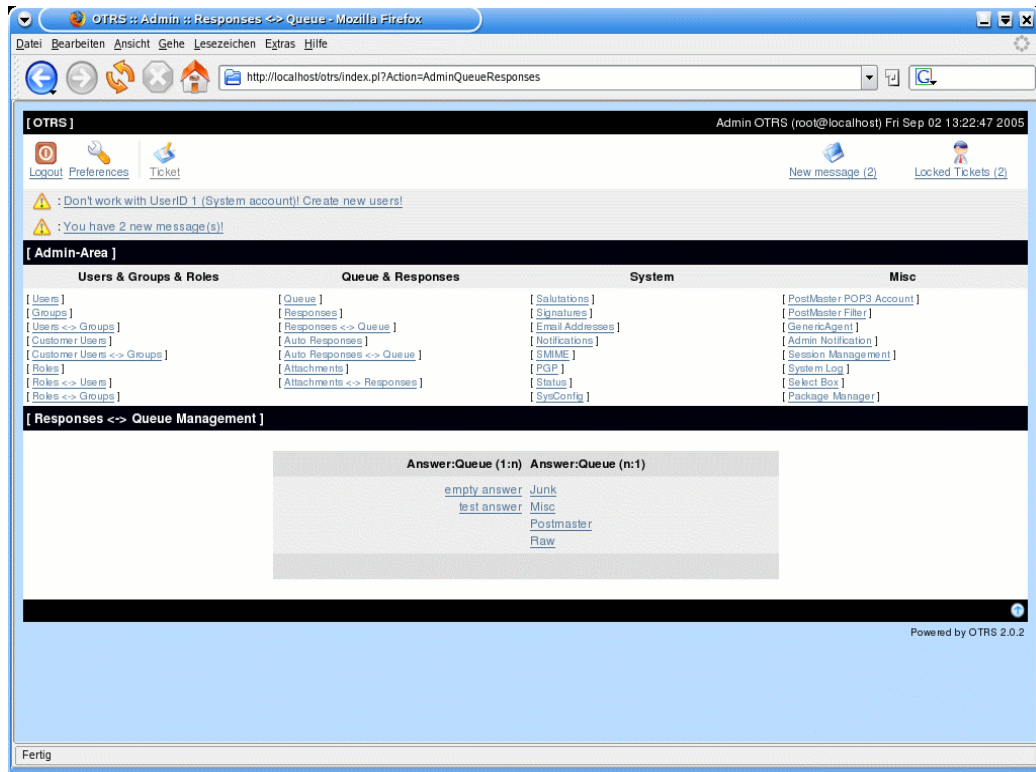
All described config settings for new queues are also valid for sub queues.

5.5. Salutations, signatures, attachments and responses

To accelerate answering tickets and to standardize the look of answers you can define responses in OTRS. A response can be linked to one or more queues, a queue can be linked to one or more responses. To make it possible to use a response quickly the different responses are displayed below of every ticket in the QueueVie or in "My Queues".

After a default installation of the system the "empty answer" response is defined for every queue. Through the "Responses" link it is possible to manage the different responses.



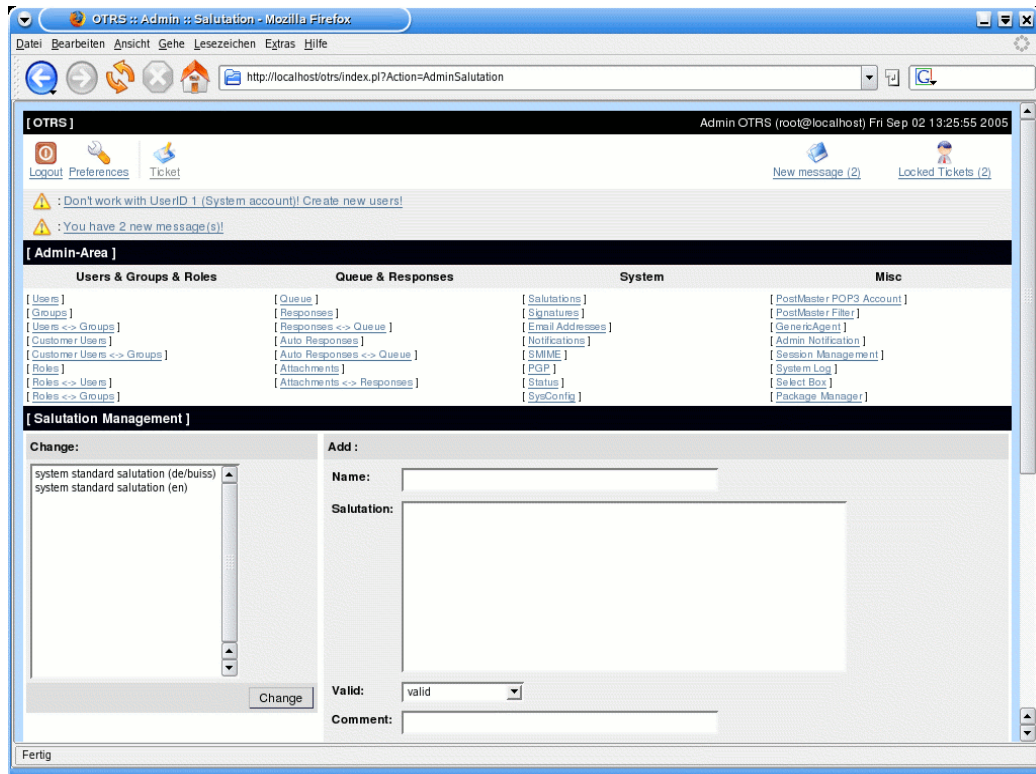


To add or remove response to one or more queues the "Responses <-> Queues" link can be used.

If you use a response for example through the QueueView you can see, that not only the text of the response and the ticket text is displayed but also a salutation and a signature. A response is assembled by different text modules. The signature and the salutation of the queue where the ticket is stored are two of the text modules of the response and also the original ticket text and, if defined, the text of the response is included. The different text modules are ordered that first the salutation is displayed, then the quoted ticket text then the text of the response and after all these text modules the signature is shown.

5.5.1. Salutations

A text module for a response is the salutation. Salutations can be linked to one or more queues as described in the section about queues. Only if a salutation is linked to a queue it is used if a ticket from this queue is answered. The "Salutations" link lets you manage the different salutations of your system.



After a default installation there are already two salutations available in the system, "system standard salutation (de/buiss)" and "system standard salutation (en)".

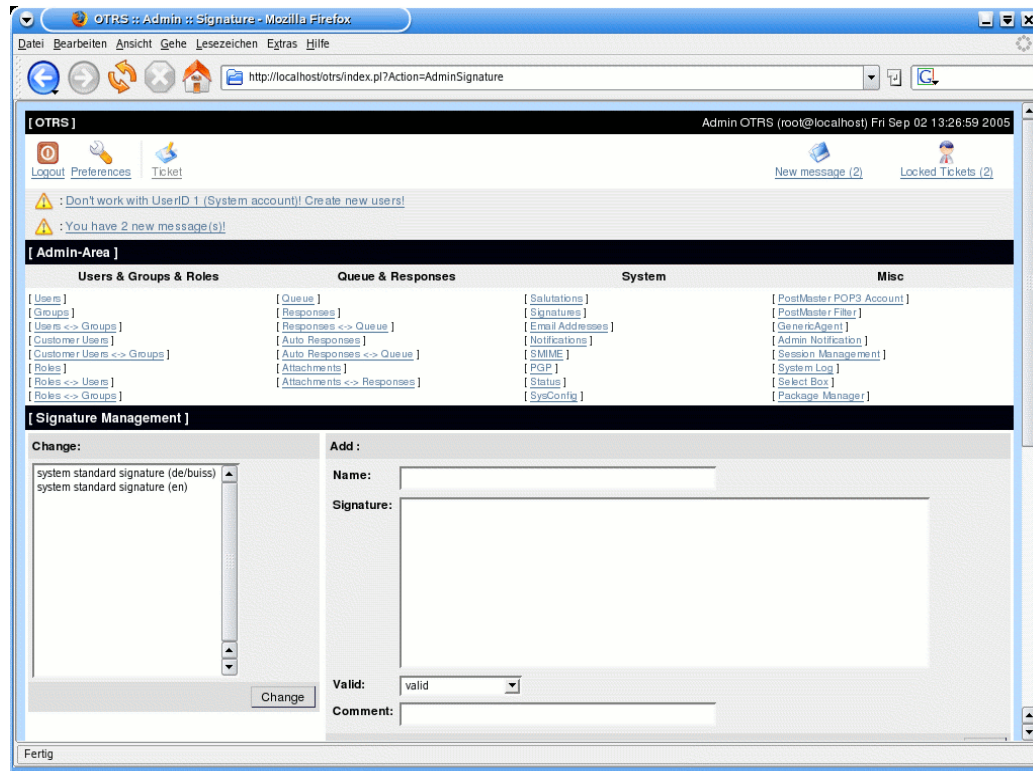
Because the content of a salutation can be created dynamicaly, for example for things that change for every ticket (e.g. the name or mail address of the sender), it is possible to use variables in salutations. The text which is saved in the variables will be placed in the response text if you reply to a ticket.

The different variables you can use in responses are listed in the lower part of the salutation screen. If you use for example the variable <OTRS_LAST_NAME> the lastname of the sender of the ticket will be included in your reply.

Note: To keep the consistency of the data for OTRS it is not possible to delete a salutation. To deactivate a salutation set it to "invalid" or "invalid-temporarily".

5.5.2. Signatures

Another text module for a response is the signature. Signatures can be linked to a queue as described in the section about the queues. Only if a signature is linked to a queue it will be included into the response text. Through "Signatures" link you can manage the signatures in your system.



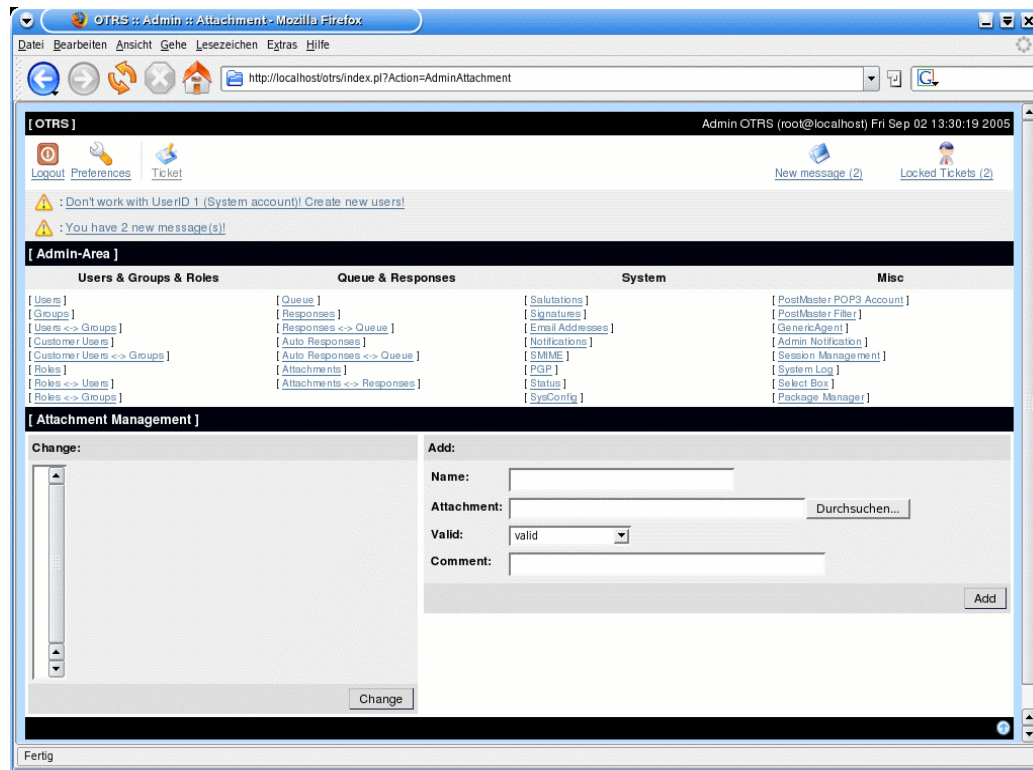
After a default installation of OTRS there are already two signatures stored in your system, "system standard signature (de/buiss)" and "system standard signature (en)".

Like salutations signatures also can contain dynamical content (e.g. the first and last name of the agent who answers the ticket), also variables replace the content of the signature text for every ticket. See the lower part of the signatures screen for the variables which can be used. If you include the variable `<OTRS_LAST_NAME>` in a signature for example, the last name of the agent who answers the ticket will be included in the signature text.

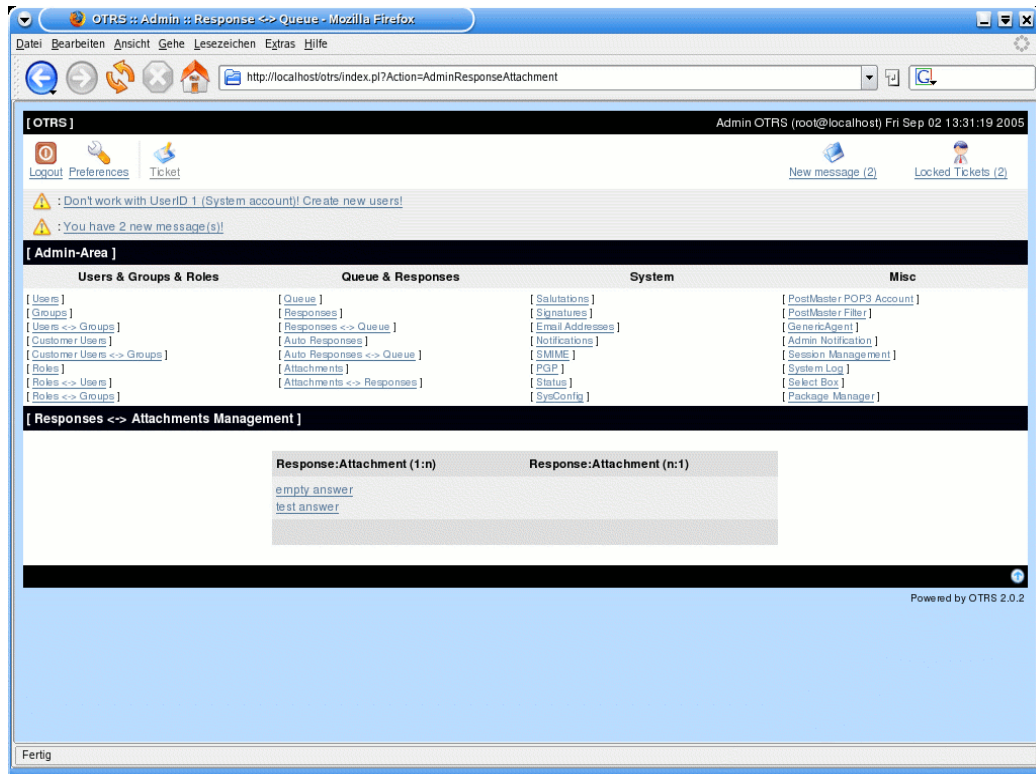
Note: To keep the consistency of the data for OTRS it is not possible to delete a signature. To deactivate a signature set it to "invalid" or "invalid-temporarily".

5.5.3. Attachments

Another optional part of a response can be one or more attachments. The attachment will be sent if the response is used, but with checkboxes it is possible to deactivate the attachment in the answer screen for tickets.



Through the "Attachment" link it is possible to load the attachments into the database of the system. If an attachment is stored it can be linked to one or more responses, just follow the "Attachment <-> Responses" link.



Note: To keep the consistency of the data for OTRS it is not possible to delete a attachment. To deactivate a attachment set it to "invalid" or "invalid-temporarily".

5.6. Auto answers

OTRS can send auto answers to customer users. Auto answers are send if special events happen, e.g. if a new ticket is created in a queue, if a followup for a ticket is received, if a ticket is closed or rejected by the system. Through the link "Auto answers" the auto answers of the system can be managed. If you create a auto answer you can select the event that should trigger the auto answer. The following system events are available after a default installation.

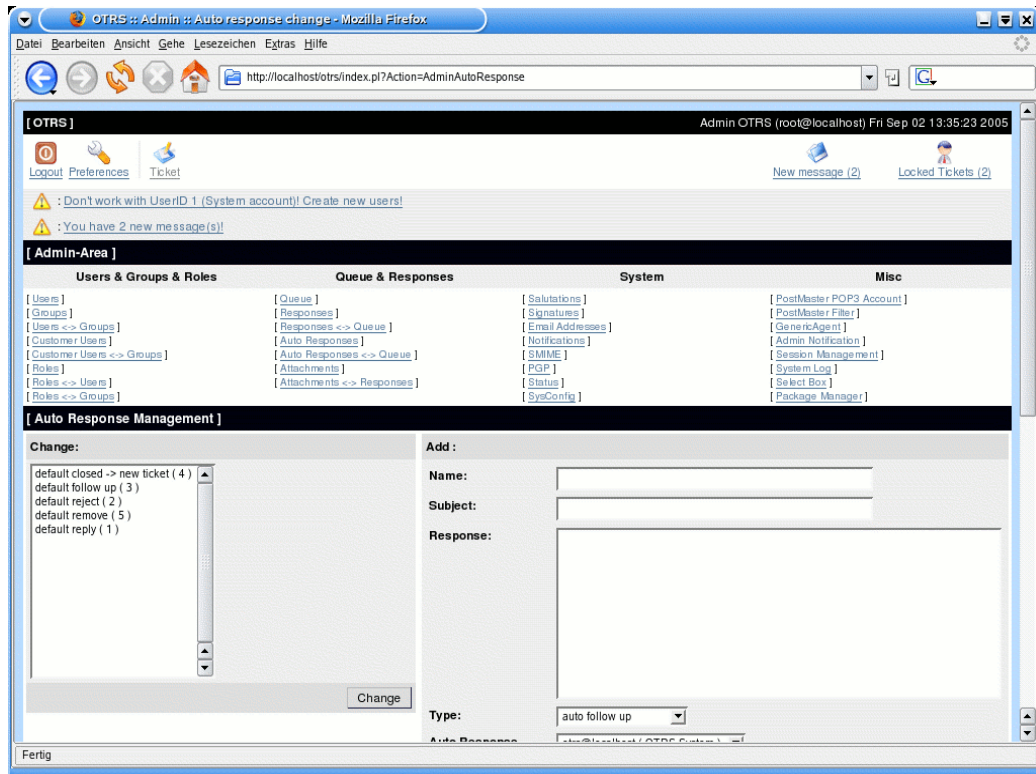
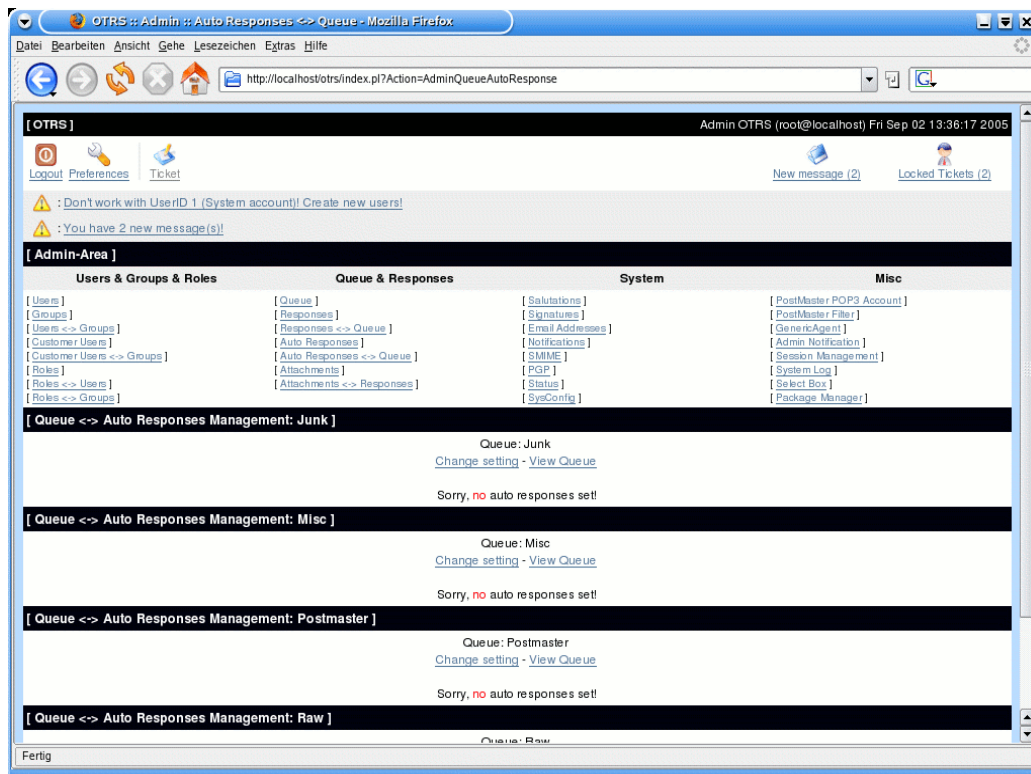


Table 5-3. Events for auto answers

Name	Description
auto reply	This event is triggered if a new ticket in a queue is created.
auto reply/new ticket	This event is triggered if a already closed ticket with a new ticketnumber is reopened, e.g. if a customer replies to such a ticket.
auto follow up	This event is triggered if a followup for a ticket is received by the system.
auto reject	This event is triggered if the system rejects a ticket automatically.
auto remove	This event is triggered if a ticket is removed by the system.

The subjectline and the text of auto answers can also be dynamically generated by variables, like for signatures or salutations. If you insert the variable <OTRS_CUSTOMER_EMAIL[5]> for example into the body of the autoanswer, the first 5 lines of the customer mailtext is inserted into the auto answer. See the lower part of the auto answer screen for more details about the variables that can be used.

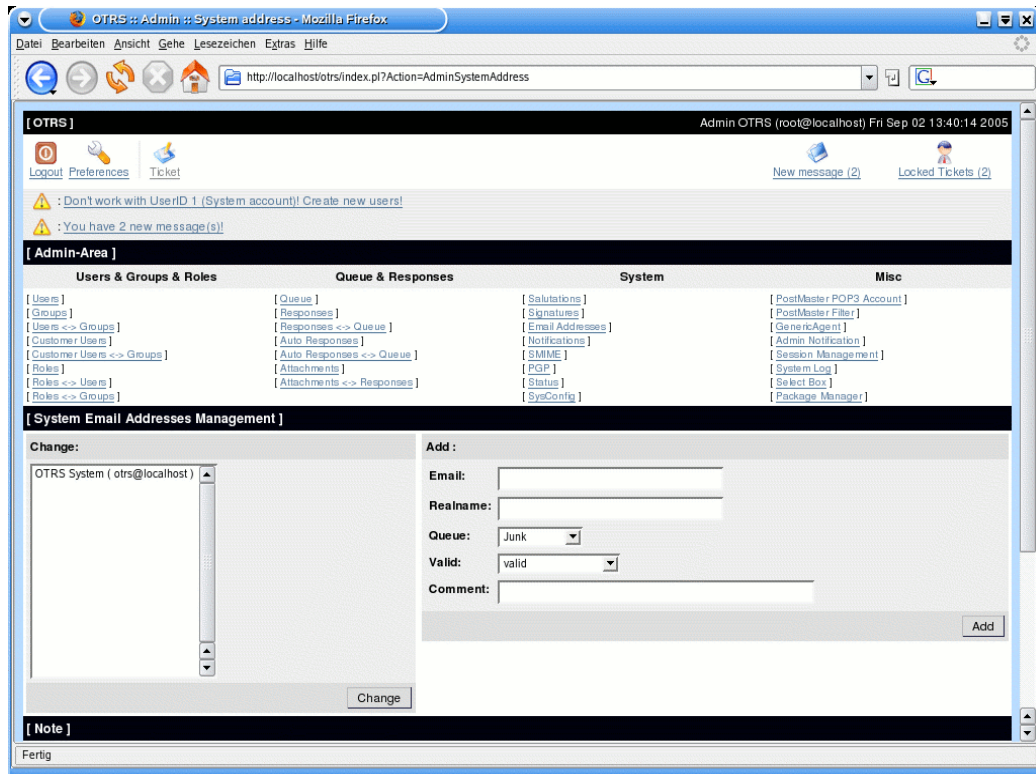
Note: To keep the consistency of the data for OTRS it is not possible to delete a auto answer. To deactivate a auto answer set it to "invalid" or "invalid-temporarily".



To add an auto answer to a queue follow the "Auto Answer <-> Queues" link in the admin area. All system events are listed for every queue and a auto answer with the same event can be selected or removed via a listbox.

5.7. Email addresses

To enable OTRS to send emails you need at least a valid email address used by the system. Because many setups need more than one mail addresses OTRS is able to work with many mail addresses at the same time. At least one mail address can be specified for a queue, but it is also possible to specify the same address for more than one queue. That means that more than one address can be used to send mails from outside to a queue, but one address has to be specified for the outgoing mails of a queue. The address, that shall be used for outgoing messages of a queue, can be set if the queue is created. Follow the "Email Addresses" link to manage all mail addresses of the system.

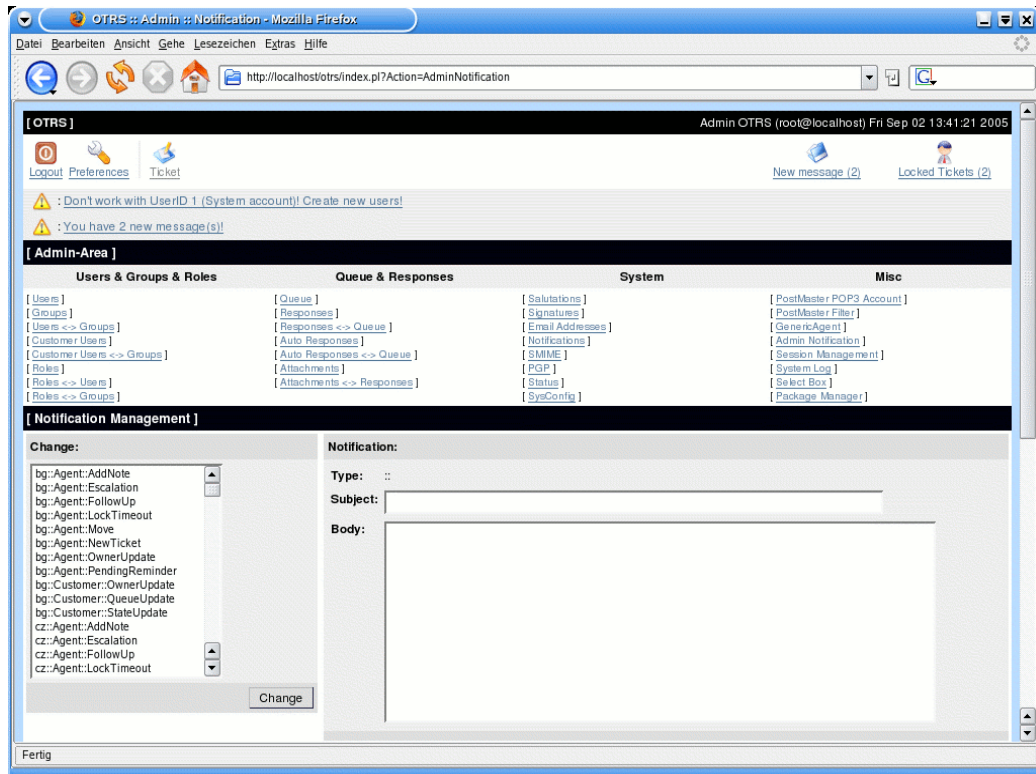


If you create a new mail address you can select the queue or sub queue that shall be linked with the new address. This link enables the system to sort incoming messages via the address in the To: field of the mail into the right queue.

Note: To keep the consistency of the data for OTRS it is not possible to delete a mail address. To deactivate a mail address set it to "invalid" or "invalid-temporarily".

5.8. Notifications

Via their preferences agents and customers can select the system events for notifications.

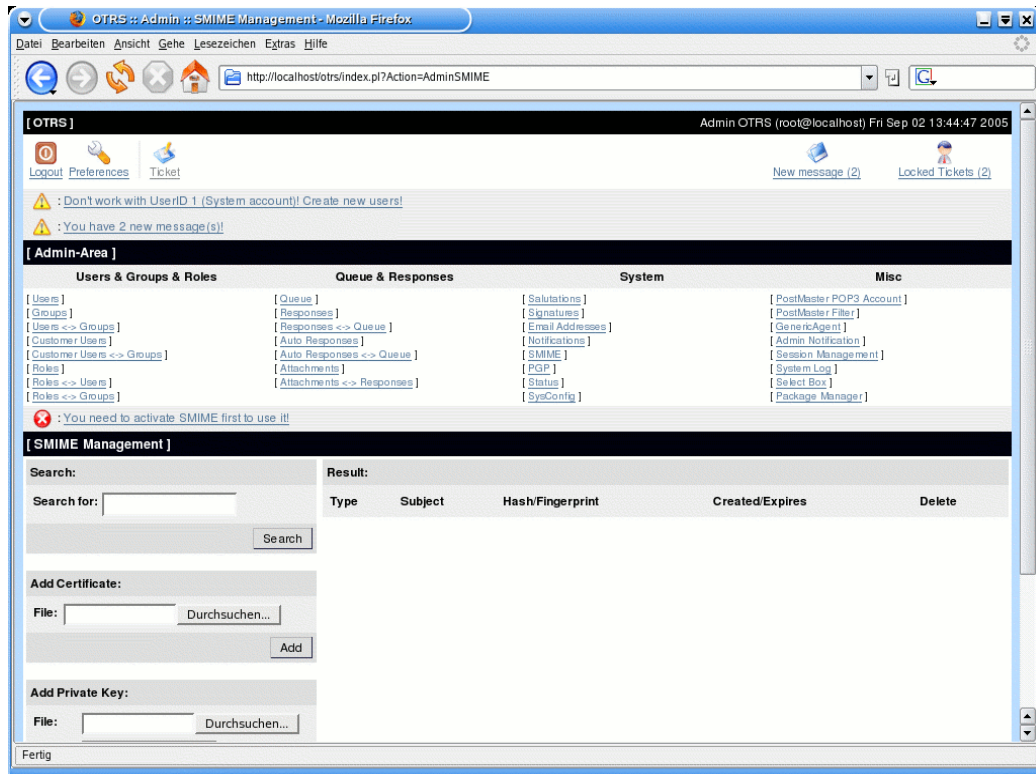


Through the "Notification" link in the adminarea you can manage the notifications of your system. You can customize the subject and the text of the notifications. Just select the notification you want to change from the listbox and load the content of the notification via the "Change" button. The name of the notification tells you for which language the notification is used, which event triggers the notification and if the notification is send to an agent or a customer.

Like signatures or salutation it is possible to create the content of a notification dynamically by using special variables. In the lower part of the notification screen you can find a list of all variables that can used for notifications.

5.9. SMIME

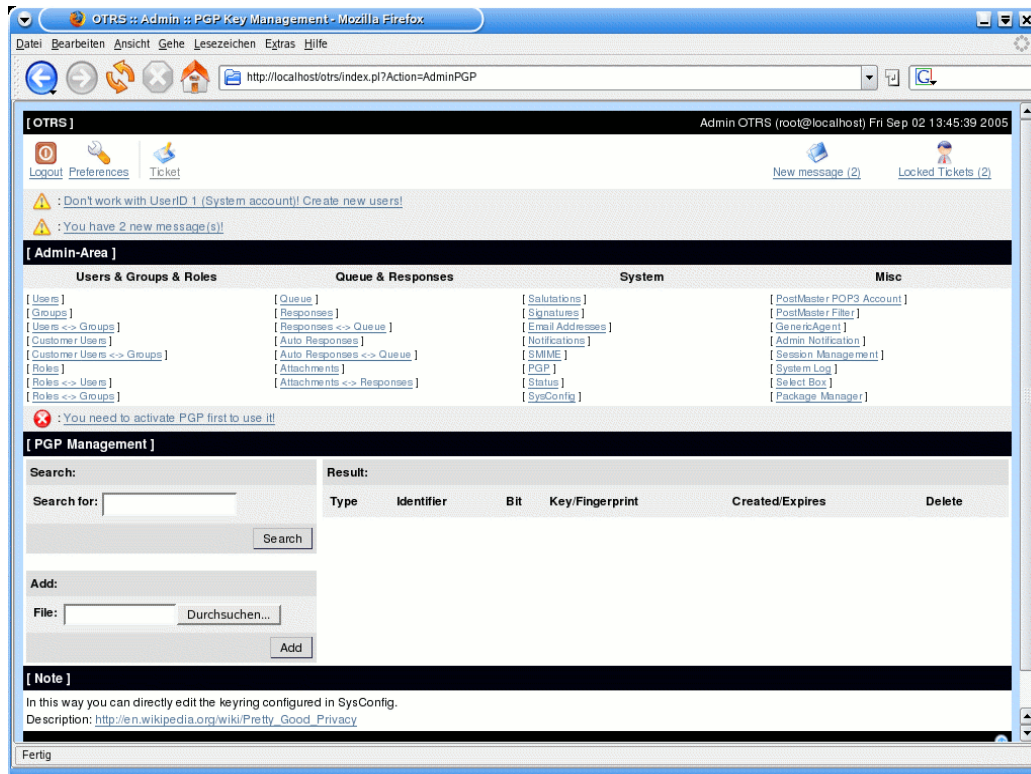
Since OTRS 2.0 it is possible to decode and encode SMIME messages. Also you can sign outgoing mails. Before this feature can be used you need to activate it and change some config parameters e.g. with the graphical configuration frontend to the needs of your operating system.



The "SMIME" link in the admin area of OTRS lets you manage your SMIME certificates. You can add or remove certificates and search through the SMIME data.

5.10. PGP

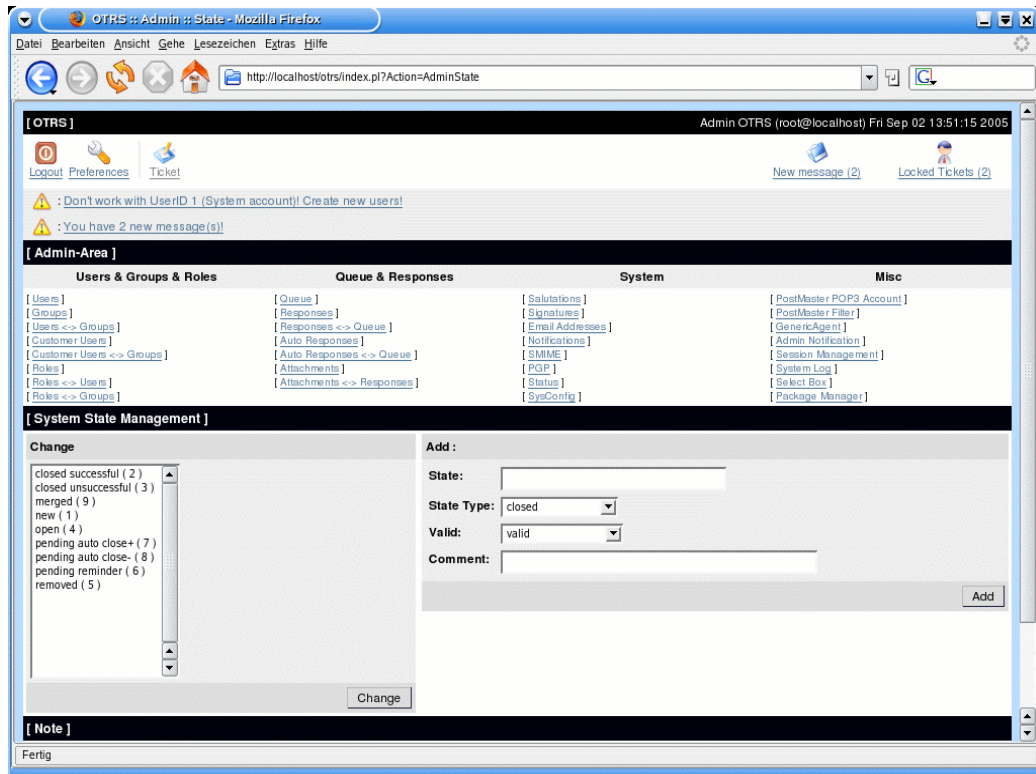
Since OTRS 2.0 it is possible to de- and encrypt messages with PGP. Also you can sign outgoing messages. Before this feature can be used you need to activate it and change some config parameters e.g. with the graphical configuration frontend to the needs of your operating system.



Through the "PGP" link in the admin area of OTRS it is possible to manage the key ring of the user who shall be used for PGP with OTRS, e.g. the local OTRS user or the web server user. It is possible to add and remove keys and signatures and you can search through all data in your key ring.

5.11. States

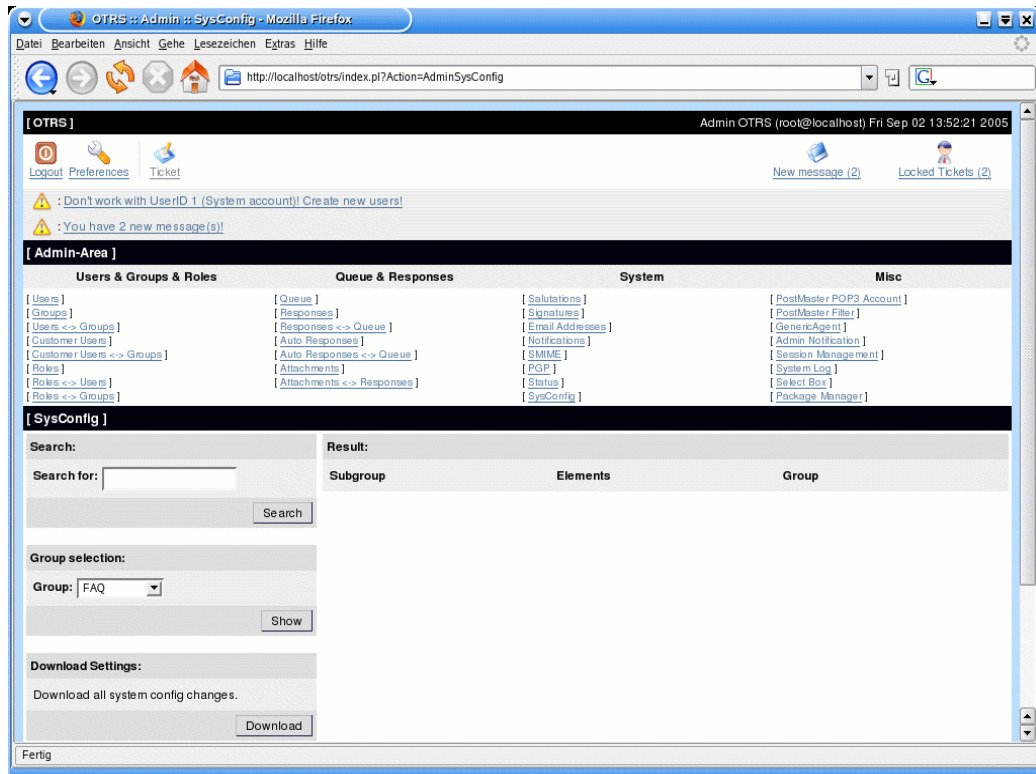
Through the "Status" link in the admin area of OTRS you can manage the different states you want to use in the ticket system.



After a default setup the states "closed successful", "closed unsuccessful", "merged", "new", "open", "pending auto close+", "pending auto close-", "pending reminder" and "removed" are already installed in the system. Every state is linked to a state type that needs to be specified if a new state is created. Per default the state types "closed", "merged", "new", "open", "pending auto", "pending reminder" and "removed".

5.12. The graphical configuration frontend (SysConfig)

Since OTRS 2.0 it is possible to change nearly all configuration parameter of the ticket system through the web interface. The new graphical configuration frontend makes it possible.

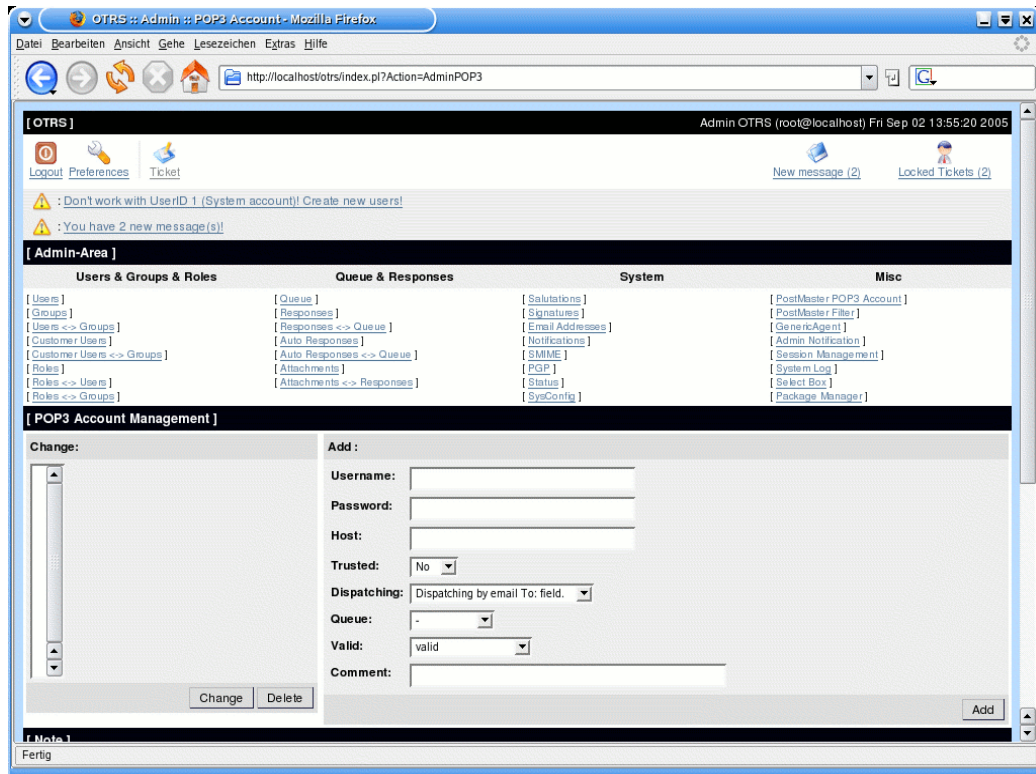


The "SysConfig" link in the admin area of OTRS loads the graphical configuration frontend. You can upload own config files for the system and backup all your current settings into a file. Almost all config parameters of the OTRS framework and the installed applications can be viewed and changed through the web interface. Because all config parameters are sorted into groups and sub groups it is possible to navigate quickly through the multitude of the parameters. Also it is possible to make a full-text search through all the config parameters.

In the chapter "Configuring the system through the web interface" the graphical configuration frontend is described more detailed.

5.13. Using POP3 accounts

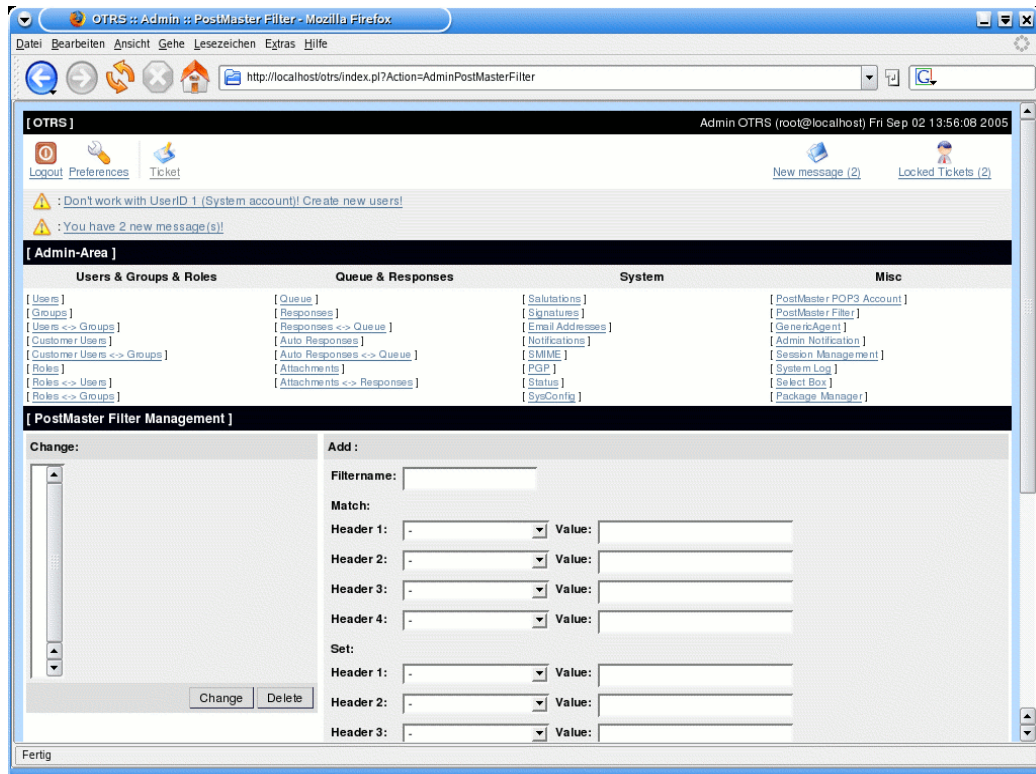
There are several possibilities to transport new emails into the ticket system. One possibility is the PostMaster.pl module that pipes the mails directly into the system. Another possibility are POP3 accounts which can be administrated through the web interface. The "PostMaster POP3 Account" link in the admin area of OTRS loads the management console for the POP3 accounts.



See the section about the PostMaster POP3 accounts for more details.

5.14. Filtering incoming messages

Because incoming messages can be sorted automatically into queues or spam mails can be moved into a specific queue, OTRS has the possibility to filter incoming messages. It does not matter, if PostMasterPOP3.pl or PostMaster.pl is used to get messages into the ticket system. Filter rules can be created through the link "PostMaster Filter" in the admin area of OTRS.



A filter rule consists of one or more filter criterias that must match if the filter rule shall be executed and of one or more actions that are executed, if the filter criterias match. You can define filter criterias for the headers or the body of an email, e.g. search for specific header entries or strings in the body, even regular expressions are allowed. All actions for a filter rule are triggered by X-OTRS headers, which are inserted if the filter criterias match. The ticket system evaluates the inserted X-OTRS headers and executes the specific actions. X-OTRS headers can be used to sort an incoming message into a specific queue, change the priority of the message or ignore the message and deliver it not to the system. The following table lists the different X-OTRS headers and their meaning.

Table 5-4. Function of the different X-OTRS-headers

Name	Possible values	Description
X-OTRS-Priority:	1 very low, 2 low, 3 normal, 4 high, 5 very high	Sets the priority of a ticket.
X-OTRS-Queue:	Name of a queue in the system.	Sets the queue where the ticket shall be sorted in. Is a queue set by a X-OTRS header all other filter rules that try to sort a ticket into a specific queue are ignored.

Name	Possible values	Description
X-OTRS-Ignore:	Yes	Is this X-OTRS header set to "Yes", the incoming message will completely be ignored and never delivered to the system.
X-OTRS-State:	new, open, closed successful, closed unsuccessful, ...	Sets the next state of the ticket.
X-OTRS-CustomerUser:	CustomerUser	Sets the customer user for the ticket.
X-OTRS-CustomerNo:	CustomerNo	Sets the customer ID for this ticket.
X-OTRS-ArticleKey(1 2 3):	Additional info key for the article.	Saves an additional info key for this article.
X-OTRS-ArticleValue(1 2 3):	Additional info value for the article.	Saves an additional info value for the article.
X-OTRS-SenderType:	agent, system, customer	Sets the type of the ticket sender.
X-OTRS-ArticleType:	email-external, email-internal, email-notification-ext, email-notification-int, phone, fax, sms, webrequest, note-internal, note-external, note-report	Sets the article type for the incoming ticket.
X-OTRS-TicketKey(1 2 ... 8):	Additional info key for the ticket.	Saves an additional info key for the ticket.
X-OTRS-TicketValue(1 2 ... 8):	Additional info value for the ticket.	Saves an additional info value for the ticket.
X-OTRS-Loop:	True	If this X-OTRS header is set no auto answer is delivered to the sender of the message (mail loop protection).

A name must be specified for every filter rule. In the section for "Match" the filter criteria can be specified. Choose via the listboxes for "Header 1", "Header 2" and so on the parts of the messages where you would like to search and specify via the input fields on the right sides the values. In the section for "Set" you can choose the actions that are triggered, if the filter rules match. You can select for "Header 1", "Header 2" and so on the X-OTRS-Header and set the values.

Example 5-1. Sort spam mails into a specific queue

A nice and useful filter rule could be to let OTRS sort mails marked for spam with Spamassassin into the "Junk" queue automatically. Spamassassin adds for every checked mail the "X-Spam-Flag" header. If the mail is marked as spam, the header is set to "Yes". So the filter criteria is "X-Spam-Flag: Yes". To create a filter rule with this criteria you can insert the name for example "spam-mails". Also choose in

the section for "Match" for "Header 1" from the listbox "X-Spam-Flag:". Insert "Yes" as value for this header. Now the filter criteria is specified. To make sure, that all spam mails get sorted into the "Junk" queue choose in the section for "Set" for "Header 1" the "X-OTRS-Queue:" entry. Specify "Junk" as value for this header. Finally add the new filter rule to activate it for the next new messages in the system.

There are additional modules, that can be used to filter incoming messages more specifically. This modules might be usefull on bigger and more complex systems.

5.15. Executing automated jobs with the GenericAgent

The GenericAgent is a tool to execute tasks automatically that must be done by a human person, a real agent, normally. The GenericAgent for example can close or move tickets, send notifications about escalated tickets, e.g.

OTRS - Admin - GenericAgent - Mozilla Firefox

http://localhost/otrs/index.pl?Action=AdminGenericAgent&Subaction=LoadProfile&Profile=Test

[GenericAgent]

Job-List

Save Job as?

Name:

Is Job Valid?

Is Job Valid:

Schedule

minutes: hours: days:

Ticket# and CustomerID

Ticket# (e. g. 10*5155 or 105658*)

CustomerID (e. g. 234321)

Customer User Login (e. g. U5150)

Fulltext-Search in Article (e. g. "Mar'in" or "Baue")

From

To

Cc

Subject

Text

Priority

1 very low

2 low

3 normal

4 high

5 very high

Queue

Junk

Misc

Postmaster

Raw

State

closed successful

closed unsuccessful

merged

new

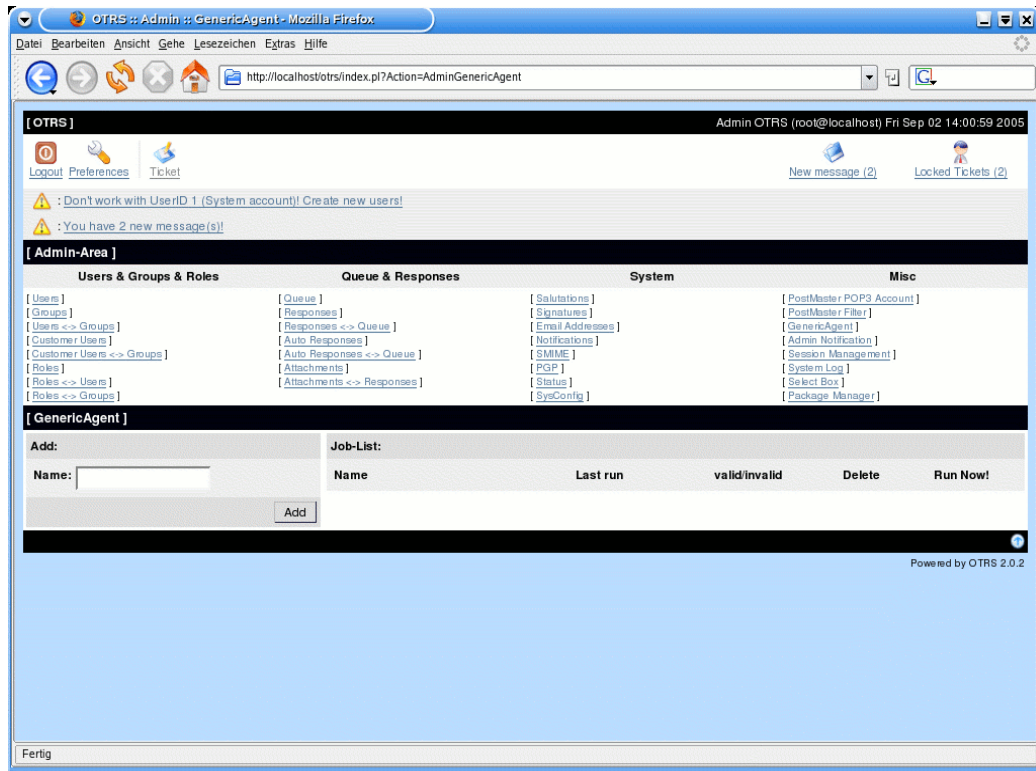
Agent/Owner

OTRS Admin (root@localhost)

Fertig

To create a new job for the GenericAgent the link "GenericAgent" in the admin area of OTRS can be used. A table with the already created jobs is displayed where jobs can be executed manually or be

removed. To create a new job a name must be specified and the "Add" button must be pressed.

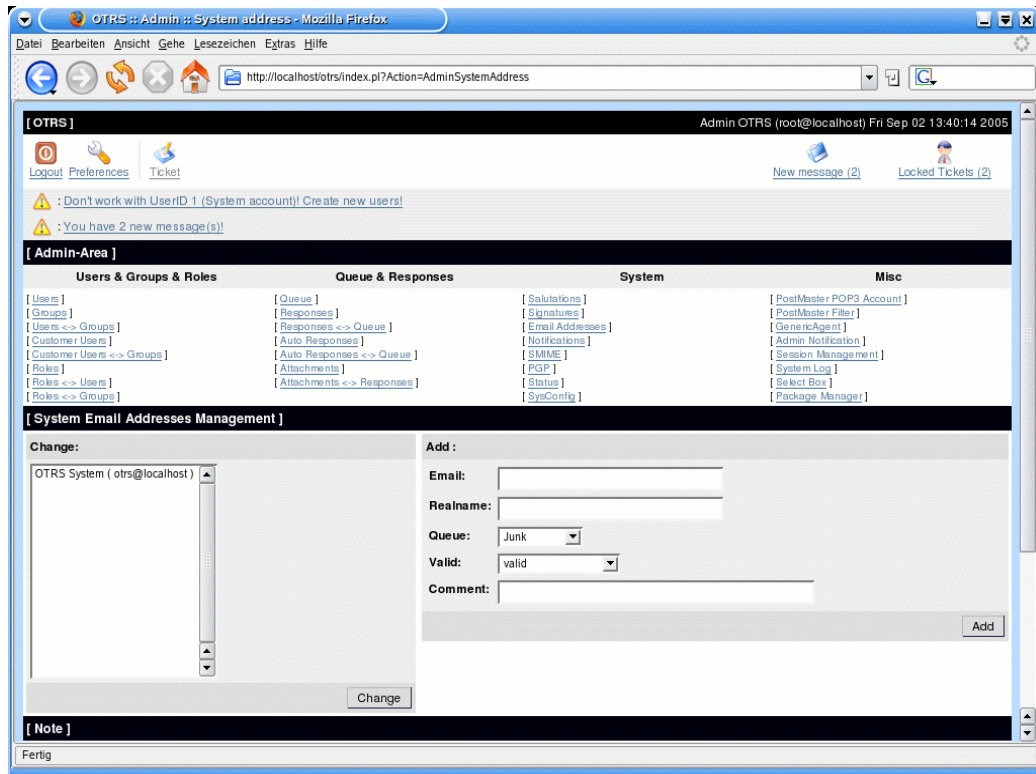


Via the screen for the creation of a new job for the GenericAgent the times can be specified when the job shall be executed. Also different criteria can be specified to select the tickets that shall be affected by the job. Also it is possible to set the new properties of the tickets which are affected by the new job.

If the creation of the job is finished all tickets are listed, that are affected by the job. This list of tickets let you control if the job works right, no changes are made to these tickets yet. The job will be really activated if it is saved into the job list.

5.16. Admin email

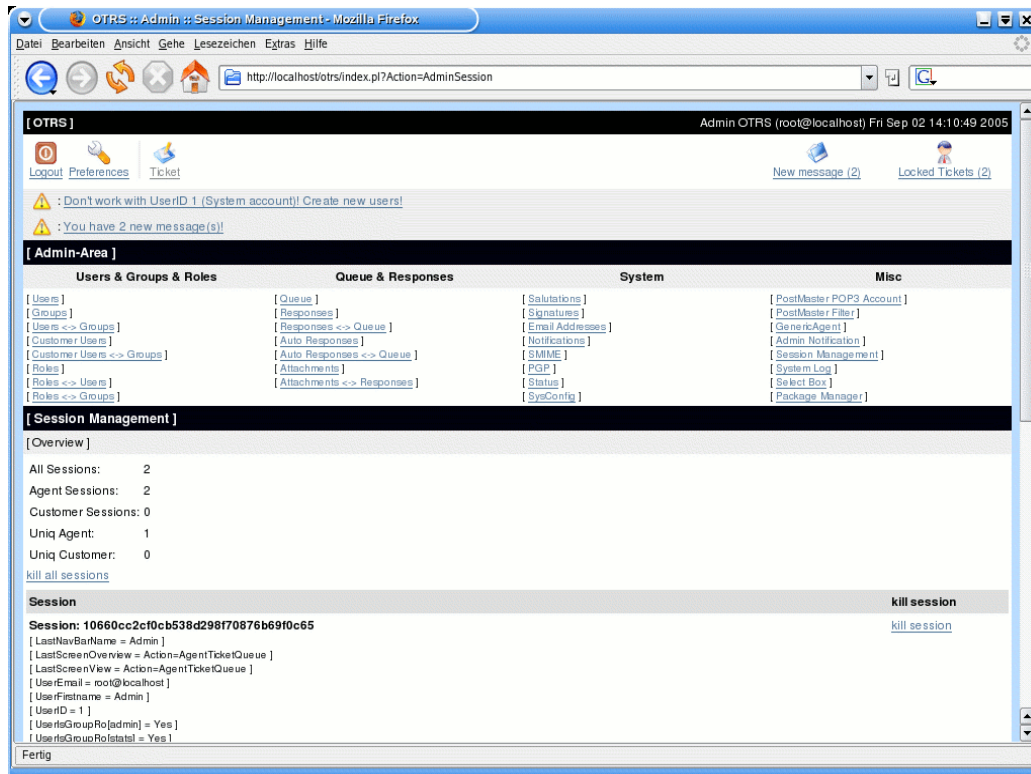
The OTRS administrator can send a message to specific users or user groups. The "Admin Notification" link opens the screen where the users and groups can be selected that should be notified by the admin.



The sender, the subject and the message text of the notification can be specified via the admin notification screen. Also the users and groups who should receive the message can be selected from the table.

5.17. Session management

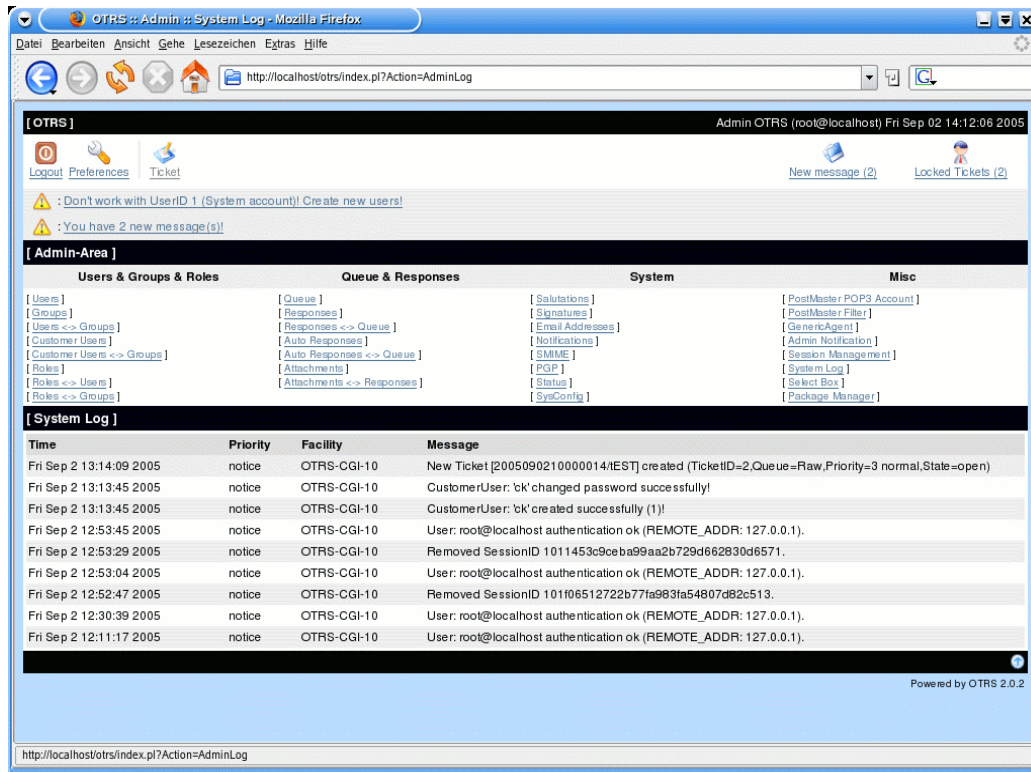
To get an overview on the currently logged in users and their session properties the "Session Management" link in the admin area can be used.



Some general information about all active sessions are displayed, e.g. how many users currently are logged in, how many agent and customer users are using the system, e.g. With the "kill all sessions" button all users sessions can be deactivated. Also detailed information for every session are available and every session can be removed separately.

5.18. System Log

The "System Log" link in the admin area of OTRS shows the last log entries of the system.



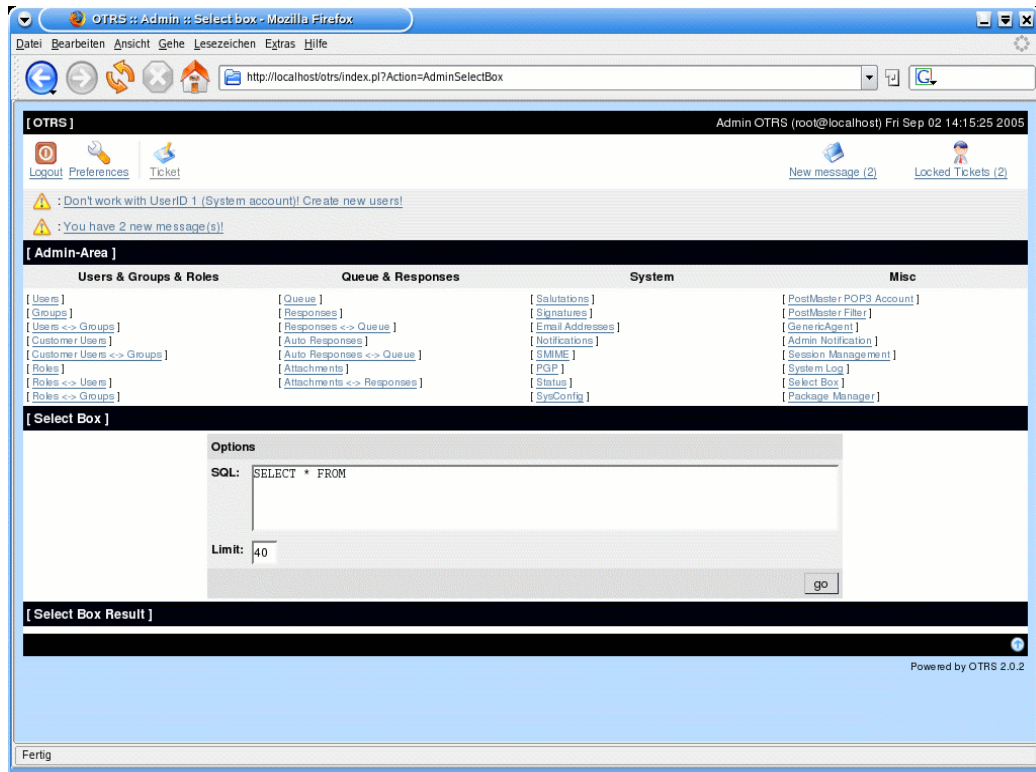
A line in the log contains a timestamp, the log priority, the system component and the log entry itself.

Note: The system logs are only available via the web interface on linux or unix systems.

The count of the shown log entries can be specified via the `LogSystemCacheSize` config parameter.

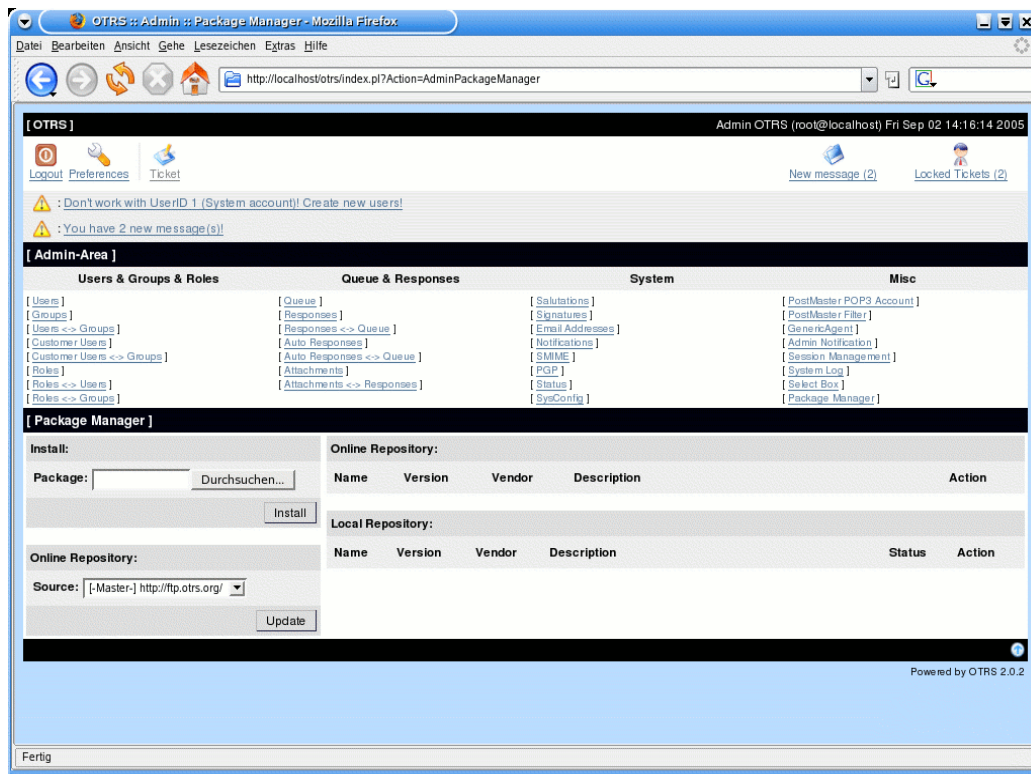
5.19. SQL queries via the select box

The "Select Box" link opens a screen that lets you query the content of the tables in the OTRS database. It is not possible to change the content of the tables, only queries are allowed.



5.20. Package manager

Since OTRS 2.0 OTRS is splitted into a central framework and some separated applications like a file manager, a web based calendar or a web mailer. To ease the handling of the separated applications the web based package manager can be used, the "Package Manager" link in the admin area of OTRS opens this module.



The package manager can only handle opm packages, other formats (rpm, deb, e.g.) are not supported. It is possible to define several installation sources. If an opm file is stored on your local hard disk you can install the package by specifying this file in the input field for "Package" via the absolute path and the file name. The "Install" button installs this package in your system and adds all necessary files, changes the database, e.g.

If always the most actual package should be used the packages can be installed via an online repository too. The latest list of all applications in the online repository can be downloaded if a server is selected via the "Source" listbox and the "Update" button is pressed. After a while on the right side of the screen all available applications are listed in the table for the online repository. You can use the "Install" button in the right column of the online repository table to integrate the module into your system. If a package depends on other packages the package manager takes care if the other needed applications are already installed. If a package has been installed successfully the application is displayed in the local repository.

If you need to remove an application you can use the "Uninstall" button in the local repository for the specific application. Dependencies to other applications are also watched during deinstallation.

Chapter 6. Configuring the system

6.1. The config files of OTRS

All config files of OTRS are stored in the directory `Kernel` or in sub directories below. There is no need to change manually any file other than `Kernel/Config.pm` as all other files will be changed when the system is upgraded. Just copy the configuration parameters from the other files into `Kernel/Config.pm` and change them to your needs, `Kernel/Config.pm` will never be touched during the upgrade process and your manual settings are saved in this file.

The file `Kernel/Config/Defaults.pm` contains the parameters of the central OTRS framework. All basic system settings like the mail configuration, database connection, default charset or standard language are located in this file. The file `Kernel/Config/Files/Ticket.pm` contains all configuration parameters for the trouble ticket system.

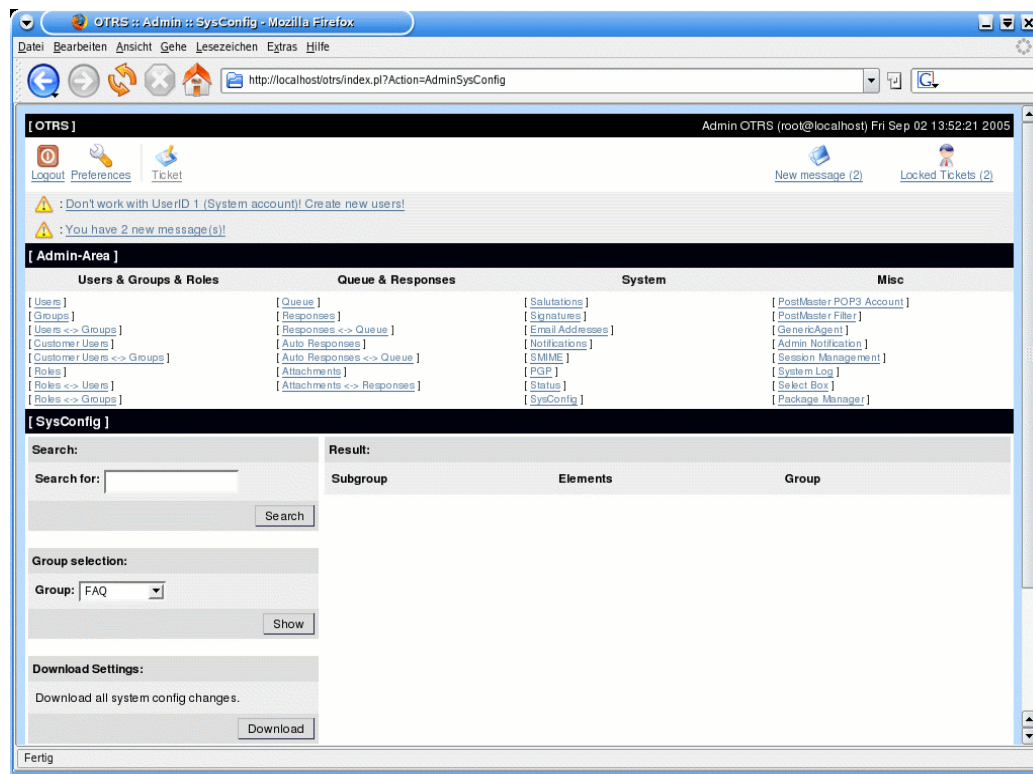
In the directory `Kernel/Config/Files` some more files are stored that are parsed when the OTRS login page is accessed. If additional applications like the web mailer or the file manager are installed, the configuration files for these applications can be found in `Kernel/Config/Files`. To keep compatibility with older OTRS versions, there is always a `.pm`- and a `.xml` file for every application and for the central framework. The `.pm` files are needed to parse all default and your own settings. The `.xml` files are used by the graphical configuration frontend which is available with OTRS 2.0 onwards and which lets you configure most of the configuration parameters of the system through your web browser. In future versions of OTRS the `.pm` files will be removed and the system will be configurable only via the `.xml` files. All parameters in `Kernel/Config/Defaults.pm` will be moved to `Kernel/Config/Files/Framework.xml` and all parameters for the trouble ticket system will be no longer stored in `Kernel/Config/Files/Ticket.pm` but in `Kernel/Config/Files/Ticket.xml`. The format of the configuration parameters will completely change to `xml`.

If the web frontend of OTRS is accessed, all `.xml` files in the `Kernel/Config/Files` directory are parsed in alphabetical order and the settings for the central framework and additional applications will be loaded. Afterwards the settings in the two files `Kernel/Config/Files/ZZZAAuto.pm` and `Kernel/Config/Files/ZZZAuto.pm` will be evaluated. Both files are used by the graphical configuration frontend and should never be changed manually. Lastly the file `Kernel/Config.pm` that contains your individual settings and manually changed configuration parameters will be parsed. Reading the configuration files in this order makes sure that your specific configuration settings are used by the system.

6.2. Configuring the system through the web interface

With OTRS 2.0 and subsequent versions nearly all configuration parameters of the central framework or additionally installed applications can be changed easily with the graphical configuration frontend. Log in

as OTRS administrator and follow the "SysConfig" link in the adminarea to execute the new configuration tool.



Because OTRS currently has over 600 configuration parameters, there are different ways to access quickly a specific configuration parameter. With the full text search all configuration parameters can be scanned for one or more keywords. The full text search not only searches through the names of the configuration parameters but the descriptions of the parameters are scanned also. This makes it possible to find a configuration parameter even if the name of the parameter is not known.

Furthermore, all configuration parameters are sorted in main groups and sub groups. The main group represents the application that the configuration parameter belongs to, e.g. "Framework" for the central OTRS framework, "Ticket" for the ticket system, "FAQ" for the FAQ system and so on. The sub groups of a main group can be accessed if the application is selected from the groups listbox and the "Show" button is pressed.

Every configuration parameter can be turned on or off via a checkbox. If the parameter is turned off, the system will ignore this parameter or the default is used. It is possible to switch a changed configuration parameter back to the system default using the "Reset" button. The "Update" button submits all changes and the system uses the new settings.

If you like to save all the changes you made to the configuration of the system, you can download a .pm file that contains all those system parameters that differ from the default. To restore your own settings it is possible to upload such a previously saved file, for example, to setup a new installed system quickly.

Note: The configuration parameters for the database connection cannot be changed via the graphical configuration frontend for security reasons. They have to be set manually in

`Kernel/Config.pm`.

Chapter 7. Sending/Receiving emails

7.1. Sending emails

7.1.1. Via Sendmail (default)

OTRS can 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.

You can configure the sendmail settings via the graphical configuration frontend (Framework::Core::Sendmail) or via 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 ';
```

7.1.2. Via SMTP relay/smarthost

OTRS can send emails via SMTP (Simple Mail Transfer Protocol / RFC 821 (<http://www.ietf.org/rfc/rfc821.txt>)). These settings will be needed generally on non-unix platforms (e.g. win32).

The sendmail settings for a smarthost can be configured either via the graphical administration frontend (Framework::Core::Sendmail) or via Kernel/Config.pm:

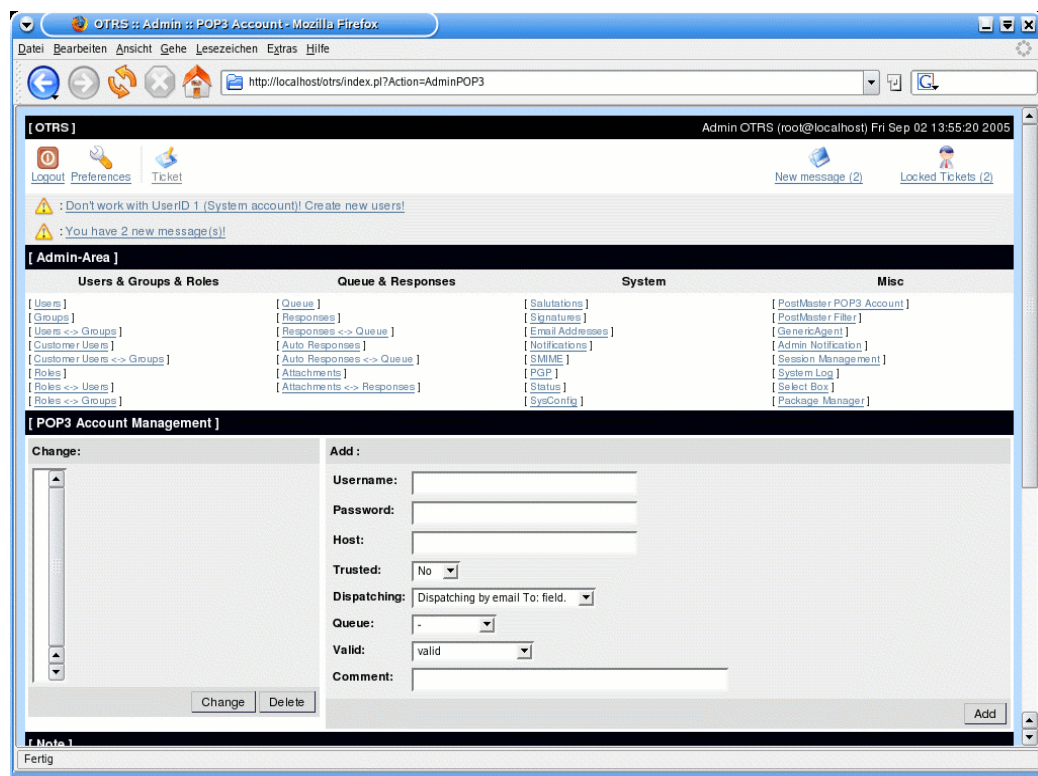
```
# SendmailModule
$Self->{"SendmailModule"} = "Kernel::System::Email::SMTP";
$Self->{"SendmailModule::Host"} = "mail.example.com";
$Self->{"SendmailModule::AuthUser"} = "";
$Self->{"SendmailModule::AuthPassword"} = "";
```

7.2. Receiving emails

7.2.1. Via POP3 accounts - the easy way (PostMasterPOP3.pl)

OTRS is able to receive emails from POP3 accounts.

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



If a new POP3 account is created then its POP3 server, username and password must be specified. If you select "Yes" for "Trusted", the X-OTRS headers are evaluated and executed if such headers are in the incoming message. Because the X-OTRS header can execute some actions in the ticket system you should only set "Trusted" to "Yes" for known senders. X-OTRS-Headers are used by the filter module in OTRS. The X-OTRS headers are explained in this table in more detailed. If you have created filter rules they are executed and used even if "Trusted" is set to "Yes".

The distribution of incoming messages can be controlled if they need to be sorted by queue or by the content of the To: field. If "Dispatching by selected queue" is selected for "Dispatching", all incoming

messages will be sorted into the specified queue. The address where the mail was sent to doesn't matter. If "Dispatching by email To: field" was selected for "Dispatching", the system checks if a queue is linked with the address in the To: field of the incoming mail. A address can be linked in the `mailaddress` management section of the admin area. If the address in the To field is linked with a queue, the new message will be sorted into the linked queue. If no link is found between the address in the To: field and queue then the message is sorted into the "Raw" queue in the system, which is the `PostmasterDefaultQueue` after a default installation.

All data for the POP3 accounts are saved in the OTRS database. The `PostMasterPOP3.pl` script, which is located in the `bin` directory of your OTRS installation, uses the settings in the database and fetches the mail. You can execute `./bin/PostMasterPOP3.pl` manually to check if all your POP3 settings are working properly. If you want to fetch your mail automatically every ten minutes, you can use a cron job and the example entries that are available in the file `var/cron/postmaster_pop3.dist`.

7.2.2. Via command line program and e.g. procmail (PostMaster.pl)

If POP3 can't be used to get the email into OTRS, the command line programm `bin/PostMaster.pl` might be a solution. `bin/PostMaster.pl` takes the mail via STDIN and pipes them directly into OTRS. That means email will be shown in your OTRS system if the MDA (mail delivery agent, e.g. procmail) executes `bin/PostMaster.pl`

To test `bin/PostMaster.pl` without MDA execute the following command:

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

If the email is shown in the QueueView then your setup is working.

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

To configure procmail for OTRS (requires a procmail configured MTA (e.g. sendmail, postfix, exim or qmail)) use the `~otrs/.procmailrc.dist` file and copy it to `.procmailrc`. 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 email sent to the local OTRS user will be piped into `bin/PostMaster.pl` and then shown in your QueueView.

7.2.3. Fetching emails via POP3 or IMAP and fetchmail for PostMaster.pl

In order to get email 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 required.

You can use the `.fetchmailrc.dist` in the home directory of OTRS and copy it to `.fetchmailrc`. Modify/change it for your needs.

Example 7-1. `.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`")!

With the `.fetchmailrc` from the example above, all email will be forwarded to the local OTRS account, if the command **fetchmail -a** is executed. Set up a cronjob with this command if you want to fetch the mails regularly.

7.2.4. Filtering/dispatching by OTRS/PostMaster modules (for more complex dispatching)

If you use the `bin/PostMaster.pl` or `bin/PostMasterPOP3.pl` method, you can insert or modify X-OTRS header entries with the PostMaster filter modules. With the X-OTRS headers the ticket system can execute some actions on incoming mails, sort them into a specific queue, change the priority order

change the customer ID for example. More information about the X-OTRS headers are available in the chapter about adding POP3 accounts in the admin area of OTRS.

There are some default filter modules:

Note: The job name (e.g. `$Self->{'PostMaster::PreFilterModule'}->{'JobName'}`) needs to be unique!

`Kernel::System::PostMaster::Filter::Match` is a default module to match on some email header (e.g. From, To, Subject, ...). It can set new email headers (e.g. X-OTRS-Ignore: yes or X-OTRS-Queue: spam) if a filter rule matches. The following example jobs can be inserted in `Kernel/Config.pm`

Example 7-2. Example jobs for the filter module `Kernel::System::PostMaster::Filter::Match`

```
# Job Name: 1-Match
# (block/ignore all spam email with From: noreply@)
$Self->{'PostMaster::PreFilterModule'}->{'1-Match'} = {
    Module => 'Kernel::System::PostMaster::Filter::Match',
    Match => {
        From => 'noreply@',
    },
    Set => {
        'X-OTRS-Ignore' => 'yes',
    },
};

# Job Name: 2-Match
# (sort emails with From: sales@example.com and Subject: **ORDER**
# into queue 'Order')
$Self->{'PostMaster::PreFilterModule'}->{'2-Match'} = {
    Module => 'Kernel::System::PostMaster::Filter::Match',
    Match => {
        To => 'sales@example.com',
        Subject => '**ORDER**',
    },
    Set => {
        'X-OTRS-Queue' => 'Order',
    },
};
```

`Kernel::System::PostMaster::Filter::CMD` is a default module to pipe the email into an external command. The output is given to STDOUT and if the result is true, then set new email header (e.g. X-OTRS-Ignore: yes or X-OTRS-Queue: spam). The following example can be used in `Kernel/Config.pm`

Example 7-3. Example job for the filter module Kernel::System::PostMaster::Filter::CMD

```
# Job Name: 5-SpamAssassin
# (SpamAssassin example setup, ignore spam emails)
$Self->{'PostMaster::PreFilterModule'}->{'5-SpamAssassin'} = {
    Module => 'Kernel::System::PostMaster::Filter::CMD',
    CMD => '/usr/bin/spamassassin | grep -i "X-Spam-Status: yes"',
    Set => {
        'X-OTRS-Ignore' => 'yes',
    },
};
```

Of course it's also possible to develop your own PostMaster filter modules.

Chapter 8. Time related functions in OTRS

8.1. Specifying the relevant times for OTRS

Some actions in OTRS are dependent on, and initiated in, relation to the actual system time. The escalation of tickets and the calculation of the escalation time for tickets depend on a proper set up of the relevant times for the ticket system. The sending of notifications for escalated tickets and for tickets that have reached the pending time is triggered in dependency to the time settings. The automatic unlock mechanism is influenced by a proper time set up too.

With the configuration parameters `TimeWorkingHours`, `TimeVacationDays` and `TimeVacationDaysOneTime` the relevant time settings for the system can be specified either via the `SysConfig` interface or directly in the `Kernel/Config.pm` file.

8.1.1. TimeWorkingHours

Set up the relevant working hours for your system in `Kernel/Config.pm` file the following way:

Example 8-1. Specifying the relevant working hours for the system

```
$Self->{'TimeWorkingHours'} = {  
    Mon => [ 8,9,10,11,12,13,14,15,16,17,18,19,20 ],  
    Tue => [ 8,9,10,11,12,13,14,15,16,17,18,19,20 ],  
    Wed => [ 8,9,10,11,12,13,14,15,16,17,18,19,20 ],  
    Thu => [ 8,9,10,11,12,13,14,15,16,17,18,19,20 ],  
    Fri => [ 8,9,10,11,12,13,14,15,16,17,18,19,20 ],  
    Sat => [ ],  
    Sun => [ ],  
};
```

Only during the specified working hours tickets can escalate, notifications for escalated and pending tickets will be send and tickets will be unlocked automatically. Furthermore only these hours influence the calculation of the escalation time and the point of time for automatic unlock.

8.1.2. TimeVacationDays

General holidays with the same date every year can be specified for the system in the `Kernel/Config.pm` file the following way:

Example 8-2. Specifying regular holidays in the system

```
$Self->{'TimeVacationDays'} = {
    1 => {
        1 => 'New Year\'s Day!',
    },
    5 => {
        1 => '1st. May',
    },
    12 => {
        24 => 'Christmas Eve',
        25 => 'Christmas Day',
        26 => 'Boxing Day',
        31 => 'New Year\'s Eve',
    },
};
```

No time related calculations or actions are done by the system On such days.

8.1.3. TimeVacationDaysOneTime

Free days or holidays that vary every year can be specified in the `Kernel/Config.pm` file the following way.

Example 8-3. Specifying irregular holidays in the system

```
$Self->{'TimeVacationDaysOneTime'} = {
    2005 => {
        12 => {
            27 => 'Annual closing',
            28 => 'Annual closing',
            29 => 'Annual closing',
            30 => 'Annual closing',
        },
    },
    2006 => {
        6 => {
            12 => 'Annual works outing',
        },
    },
};
```

No time related calculations or actions are done by the system On such days.

8.2. Automated unlocking of tickets

Locked tickets can be unlocked automatically by the system. This feature might be useful if an agent has locked tickets that need to be processed, but the agent can't work on this tickets because the agent is on holiday, for example. The automated unlock feature unlocks tickets after a given time to ensure that no locked tickets will be forgotten and other agents can process these tickets.

The unlock time for tickets can be specified in the `settings` for every queue. The module `bin/UnlockTickets.pl`, which should be executed periodically via a cron job, performs the automated unlocking of tickets. Tickets are unlocked only during the hours set by the `TimeWorkingHours` configuration parameter. During days specified in `TimeVacationDays` and `TimeVacationDaysOneTime` no tickets are unlocked automatically.

Notifications on unlocked tickets are send out only to those agents that have the queue with the unlocked tickets set in "My queues" and that have activated the notification on unlocked tickets in their personal preferences .

8.3. Pending tickets

With OTRS you can create pending tickets. The system can send out a notification at a given time and remind an agent of a locked ticket. This feature might be useful to get reminded about contacting a customer sometime in the future as the customer is not reachable at the moment, for example.

Reminders on pending tickets are only send out during hours specified with the `TimeWorkingHours` configuration parameter. The `bin/PendingJobs.pl` module, which should be executed periodically via a cron job, triggers the delivery of the pending reminders.

8.4. Ticket escalation

With OTRS it is possible to let tickets escalate. If a ticket is escalated, no other tickets are displayed for the queue containing the escalated ticket. The escalated ticket has to be locked first. The escalation of tickets ensures that tickets must be considered after they have reached a given age.

The age after which a ticket escalates can be specified in the `settings` for every queue. With a job for the `GenericAgent` notifications on escalated tickets can be delivered to agents that have the queue with the escalated ticket in "My queues" enabled and that have activated notifications on escalated tickets in their personal preferences .

Example 8-4. GenericAgent job to send notification on escalated tickets

The file `Kernel/Config/GenericAgent.pm` already contains an example job for notification delivery to agents on escalated tickets. The jobs in this file are processed by the `GenericAgent`, which is executed periodically by a cron job. Just open the file and remove the comments ("`#`") for the following lines:

```
%Jobs = (
  # --
  # [name of job] -> send escalation notifications
  # --
  'send escalation notifications' => {
    Escalation => 1,
    # new ticket properties
    New => {
      Module => 'Kernel::System::GenericAgent::NotifyAgentGroupOfCustomQueue',
    },
  },
  # insert your jobs (see Kernel/Config/GenericAgent.pm.examples)
);
```

If a new ticket is stored in a queue that has set a value for escalation time, the positive value for the set escalation time is shown first. The displayed value for the escalation time does not change if the system time is not in the hours specified for `TimeWorkingHours` or if today is a day specified in `TimeVacationDays` or `TimeVacationDaysOneTime`.

If the system is in a time window which is relevant for the calculation for time specific events, the escalation time is counted down. If the value 0 is reached then the ticket escalates. When the value for the escalation time becomes negative, the ticket has exceeded the point of time for the escalation. During the next run of the `GenericAgent` the job for the notification about escalated tickets is executed and the messages will be sent out to the agents. The escalated ticket blocks the view of all other tickets in the queue and must be processed. Even if the escalated ticket is locked and processed the displayed value for the escalation time stays in the negative range. This behaviour does not change as long as the ticket is in an open state (open, new, pending, etc.). The value for the escalation time only gets reset if the state changes to closed. If the same ticket is reopened, for example, by a follow up ticket, the count down for the escalation time starts again with the positive value set for the queue in which the ticket is stored.

Thus a ticket only can escalate if it is not locked and is in an open state. If the ticket is locked and not yet closed, the escalation time is counted down and the ticket escalates if the time value is zero. If the state was changed to closed and if the ticket is reopened by a follow-up request, the escalation process described above restarts.

Chapter 9. Using external backends

9.1. Customer data

OTRS is able to handle different customer data, e.g. login, email, phone number. These information can be displayed in the customer interface and they are useable in the customer frontend. Also these data are used for the authentication of customer users.

The used / displayed customer data are freely configurable, but without the following data OTRS won't work, this information is always needed for the authentication of customers.

Necessary data needed for customer authentication

- User login
- Email address
- Customer ID

Use the following configuration parameters in your `Kernel/Config.pm` file, if you want to display customer information in your agent interface.

```
# Ticket::Frontend::CustomerInfo*
# (show customer user info on Compose (Phone and Email), Zoom and
# Queue view)
$Self->{'Ticket::Frontend::CustomerInfoCompose'} = 1;
$Self->{'Ticket::Frontend::CustomerInfoZoom'} = 1;
$Self->{'Ticket::Frontend::CustomerInfoQueue'} = 0;
```

9.2. Customer user backend

You can use two types of customer backends, DB and LDAP. If you already have another customer backend (e.g. SAP) it is ofcourse possible to write a module that uses this type of backend.

9.2.1. Database (Default)

Example 9-1. Configuring a DB customer backend

This example shows the configuration of a DB customer backend which uses customer data stored in the OTRS database.

```
# CustomerUser
# (customer user database backend and settings)
$Self->{CustomerUser} = {
    Name => 'Datenbank Quelle',
    Module => 'Kernel::System::CustomerUser::DB',
    Params => {
        # if you want to use an external database, add the
        # required settings
        DSN => 'DBI:odbc:yourdsn',
        DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',
        User => "",
        Password => "",
        Table => 'customer_user',
    },
    # customer uniq id
    CustomerKey = 'login',
    # customer #
    CustomerID = 'customer_id',
    CustomerValid = 'valid_id',
    CustomerUserListFields => ['first_name', 'last_name', 'email'],
    CustomerUserSearchFields => ['login', 'last_name', 'customer_id'],
    CustomerUserSearchPrefix => "",
    CustomerUserSearchSuffix => '*',
    CustomerUserSearchListLimit => 250,
    CustomerUserPostMasterSearchFields => ['email'],
    CustomerUserNameFields => ['salutation', 'first_name', 'last_name'],
    CustomerUserEmailUniqCheck => 1,
    # show now own tickets in customer panel, CompanyTickets
    # CustomerUserExcludePrimaryCustomerID => 0,
    # generate auto logins
    # AutoLoginCreation => 0,
    # AutoLoginCreationPrefix => 'auto',
    # admin can change customer preferences
    # AdminSetPreferences => 1,
    # just a read only source
    # ReadOnly => 1,
    Map => [
        # note: Login, Email and CustomerID needed!
        # var, frontend, storage, shown (1=always,2=lite), required, storage-type, http
        [ 'UserSalutation', 'Salutation', 'salutation', 1, 0, 'var', "", 0 ],
        [ 'UserFirstname', 'Firstname', 'first_name', 1, 1, 'var', "", 0 ],
        [ 'UserLastname', 'Lastname', 'last_name', 1, 1, 'var', "", 0 ],
        [ 'UserLogin', 'Username', 'login', 1, 1, 'var', "", 0 ],
        [ 'UserPassword', 'Password', 'pw', 0, 1, 'var', "", 0 ],
    ]
}
```

```

[ 'UserEmail',      'Email',      'email',      0, 1, 'var', "", 0 ],
# [ 'UserEmail',      'Email',      'email',      1, 1,
# 'var', '$Env{"CGIHandle"}?Action=AgentTicketCompose&ResponseID=1&TicketID=$Data
[ 'UserCustomerID', 'CustomerID', 'customer_id', 0, 1, 'var', "", 0 ],
# [ 'UserCustomerIDs', 'CustomerIDs', 'customer_ids', 1, 0, 'var', "", 0 ],
[ 'UserComment',      'Comment',      'comments',      1, 0, 'var', "", 0 ],
[ 'ValidID',          'Valid',          'valid_id',      0, 1, 'int', "", 0 ],
],
# default selections
Selections => {
  UserSalutation => {
    'Mr.' => 'Mr.',
    'Mrs.' => 'Mrs.',
  },
},
};

```

If you want to customize the customer data, change the table columns in the customer_user table in the OTRS database or add new columns. In the following example a new field for phone number is added.

```

linux:~# mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 116 to server version: 5.0.18-Debian_7-log

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

mysql> use otrs;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> ALTER TABLE customer_user ADD phone VARCHAR (250);
Query OK, 1 rows affected (0.01 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql> quit
Bye
linux:~#

```

Now add the new column to the MAP array in Kernel/Config.pm:

```

# var, frontend, storage, shown (1=always,2=lite), required, storage-type, http-link, r
[...]
[ 'UserPhone',      'Phone',      'phone',      0, 1, 'var', "", 0 ],

```

Ofcourse it is possible to edit all these customer information via the customer area in the agent interface.

9.2.1.1. Customer with multible IDs (Company tickets)

It is possible to assign more than one customer ID to a customer. This can be usefull if a customer must access tickets of other customers, e.g. a supervisor wants to watch the tickets of his assistants. If a customer can access the tickets of another customer user, the company ticket feature of OTRS is used. Company tickets can be accessed via the "Company Ticket" link in the customer panel.

To use company tickets a new column has to be added to the customer_user table in the OTRS database. In this new column the IDs of the customers are stored that tickets need to be accessed.

```
linux:~# mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 124 to server version: 5.0.18-Debian_7-log

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

mysql> use otrs;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> ALTER TABLE customer_user ADD customer_ids VARCHAR (250);
Query OK, 1 rows affected (0.02 sec)
Records: 1  Duplicates: 0  Warnings: 0

mysql> quit
Bye
linux:~#
```

Now the new column has to be added to the MAP array in Kernel/Config.pm:

```
# var, frontend, storage, shown (1=always,2=lite), required, storage-type, http-link, r
[...]
[ 'UserCustomerIDs', 'CustomerIDs', 'customer_ids', 1, 0, 'var', "", 0 ],
```

The new column for the multi customer IDs can be edited via the agent interface in the section for the customer user management now.

To ensure that one customer can access the tickets of other customer users add the IDs of this other users into the new field for the multiple customer IDs. Each ID has to be separated by a semicolon.

Example 9-2. Using company tickets with a DB backend

The customers A, B and C exist in your system and A wants to have access to the tickets of B and C via the customer panel. B and C should have no access to the tickets of other users.

To realize this setup change the `customer_user` table and the mapping in `Kernel/Config.pm` like described above. Then load the settings for customer A via the customer area in the agent interface or via the admin area. If the settings are displayed add into the field for CustomerIDs the values "B;C".

9.2.2. LDAP

If you have a LDAP directory with your customer data you can use it as customer backend with OTRS.

Example 9-3. Configuring a LDAP customer backend

This is an example for a customer backend that has stored all customer information in a LDAP directory.

```
# CustomerUser
# (customer user ldap backend and settings)
$Self->{CustomerUser} = {
    Name => 'LDAP Datenquelle',
    Module => 'Kernel::System::CustomerUser::LDAP',
    Params => {
        # ldap host
        Host => 'bay.csu Hayward.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 => "",
        # in case you want to add always one filter to each ldap query, use
        # this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter => '(objectclass=*)'
        AlwaysFilter => "",
    },
};
```

```

# if your frontend is e. g. iso-8859-1 and the charset of your
# ldap server is utf-8, use this options (if not, ignore it)
# SourceCharset => 'utf-8',
# DestCharset => 'iso-8859-1',
# Net::LDAP new params (if needed - for more info see perldoc Net::LDAP)
Params => {
    port => 389,
    timeout => 120,
    async => 0,
    version => 3,
},
},
# customer uniq id
CustomerKey => 'uid',
# customer #
CustomerID => 'mail',
CustomerUserListFields => ['cn', 'mail'],
CustomerUserSearchFields => ['uid', 'cn', 'mail'],
CustomerUserSearchPrefix => "",
CustomerUserSearchSuffix => '*',
CustomerUserSearchListLimit => 250,
CustomerUserPostMasterSearchFields => ['mail'],
CustomerUserNameFields => ['givenname', 'sn'],
# show now own tickets in customer panel, CompanyTickets
CustomerUserExcludePrimaryCustomerID => 0,
# add a ldap filter for valid users (expert setting)
# CustomerUserValidFilter => '(! (description=gesperrrt))',
# admin can't change customer preferences
AdminSetPreferences => 0,
Map => [
    # note: Login, Email and CustomerID needed!
    # var, frontend, storage, shown (1=always,2=lite), required, storage-type, http
    [ 'UserSalutation', 'Title', 'title', 1, 0, 'var', "", 0 ],
    [ 'UserFirstname', 'Firstname', 'givenname', 1, 1, 'var', "", 0 ],
    [ 'UserLastname', 'Lastname', 'sn', 1, 1, 'var', "", 0 ],
    [ 'UserLogin', 'Username', 'uid', 1, 1, 'var', "", 0 ],
    [ 'UserEmail', 'Email', 'mail', 1, 1, 'var', "", 0 ],
    [ 'UserCustomerID', 'CustomerID', 'mail', 0, 1, 'var', "", 0 ],
    # [ 'UserCustomerIDs', 'CustomerIDs', 'second_customer_ids', 1, 0, 'var', "", 0 ]
    [ 'UserPhone', 'Phone', 'telephonenumber', 1, 0, 'var', "", 0 ],
    [ 'UserAddress', 'Address', 'postaladdress', 1, 0, 'var', "", 0 ],
    [ 'UserComment', 'Comment', 'description', 1, 0, 'var', "", 0 ],
],
};

```

If additional customer information are stored in your LDAP directory and if you like to use this information with OTRS also, just expand the MAP array in `Kernel/Config.pm` with the entries for this data.

```
# var, frontend, storage, shown (1=always,2=lite), required, storage-type, http-link, r
[...]
[ 'UserPhone',      'Phone',      'telephonenumber', 1, 0, 'var', "", 0 ],
```

9.2.2.1. Customer with multiple IDs (Company tickets)

It is possible to assign more than one customer ID to a customer. This can be usefull if a customer must access tickets of other customers, e.g. a supervisor wants to watch the tickets of his assistants. If a customer can access the tickets of another customer user, the company ticket feature of OTRS is used. Company tickets can be accessed via the "Company Ticket" link in the customer panel.

Du use company tickets a new field has to be added to the LDAP directory that can contain the entries with the IDs for the customers that tickets should be accessible for a customer user.

If the new field in the LDAP directory has been created the new entry has to be added to the MAP array in `Kernel/Config.pm`:

```
# var, frontend, storage, shown (1=always,2=lite), required, storage-type, http
[...]
[ 'UserCustomerIDs', 'CustomerIDs', 'customer_ids', 1, 0, 'var', "", 0 ],
```

The field for the multiple customer IDs has to be edited directly in the LDAP directory, OTRS can only read from LDAP.

To ensure the access for a customer to the tickets of other customers add the customer IDs of the customers that tickets should be accessed to the new field in your LDAP directory. Each ID has to be seperated by a semicolon.

Example 9-4. Using Company tickets with a LDAP backend

The customers A, B and C exists in your system and A wants to have access to the tickets of B and C via the customer panel. B and C should have no access to tickets of other users.

To realize this setup change the LDAP directory and the mapping in `Kernel/Config.pm` like described above. Then add into the field for CustomerIDs the values "B;C;" for customer A in your LDAP directory.

9.2.3. Use more than one customer backend with OTRS

If more than one customer backend with different customer data should be used with OTRS (e.g. a LDAP and a DB backend), the CustomerUser config parameter has to be expanded with a number, e.g. "CustomerUser1", "CustomerUser2".

Example 9-5. Using more than one customer backend with OTRS

The following configuration example shows parallel usage of an LDAP and DB customer backend with OTRS.

```
# 1. Customer user backend: DB
# (customer user database backend and settings)
$Self->{CustomerUser1} = {
    Name => 'Datenbank Quelle',
    Module => 'Kernel::System::CustomerUser::DB',
    Params => {
        # if you want to use an external database, add the
        # required settings
        DSN => 'DBI:odbc:yourdsn',
        DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',
        User => "",
        Password => "",
        Table => 'customer_user',
    },
    # customer uniq id
    CustomerKey = 'login',
    # customer #
    CustomerID = 'customer_id',
    CustomerValid = 'valid_id',
    CustomerUserListFields => ['first_name', 'last_name', 'email'],
    CustomerUserSearchFields => ['login', 'last_name', 'customer_id'],
    CustomerUserSearchPrefix => "",
    CustomerUserSearchSuffix => '*',
    CustomerUserSearchListLimit => 250,
    CustomerUserPostMasterSearchFields => ['email'],
    CustomerUserNameFields => ['salutation', 'first_name', 'last_name'],
    CustomerUserEmailUniqCheck => 1,
    # show now own tickets in customer panel, CompanyTickets
    CustomerUserExcludePrimaryCustomerID => 0,
    # generate auto logins
    AutoLoginCreation => 0,
    AutoLoginCreationPrefix => 'auto',
    # admin can change customer preferences
    AdminSetPreferences => 1,
    # just a read only source
    ReadOnly => 1,
    Map => [
        # note: Login, Email and CustomerID needed!
        # var, frontend, storage, shown (1=always,2=lite), required, storage-type, http
```



```

[ 'UserSalutation', 'Salutation', 'salutation', 1, 0, 'var', "", 0 ],
[ 'UserFirstname', 'Firstname', 'first_name', 1, 1, 'var', "", 0 ],
[ 'UserLastname', 'Lastname', 'last_name', 1, 1, 'var', "", 0 ],
[ 'UserLogin', 'Username', 'login', 1, 1, 'var', "", 0 ],
[ 'UserPassword', 'Password', 'pw', 0, 1, 'var', "", 0 ],
[ 'UserEmail', 'Email', 'email', 0, 1, 'var', "", 0 ],
# [ 'UserEmail', 'Email', 'email', 1, 1,
# 'var', '$Env{"CGIHandle"}?Action=AgentTicketCompose&ResponseID=1&TicketID=$Data
[ 'UserCustomerID', 'CustomerID', 'customer_id', 0, 1, 'var', "", 0 ],
# [ 'UserCustomerIDs', 'CustomerIDs', 'customer_ids', 1, 0, 'var', "", 0 ],
[ 'UserComment', 'Comment', 'comments', 1, 0, 'var', "", 0 ],
[ 'ValidID', 'Valid', 'valid_id', 0, 1, 'int', "", 0 ],
],
# default selections
Selections => {
    UserSalutation => {
        'Mr.' => 'Mr.',
        'Mrs.' => 'Mrs.',
    },
},
};

# 2. Customer user backend: LDAP
# (customer user ldap backend and settings)
$Self->{CustomerUser2} = {
    Name => 'LDAP Datenquelle',
    Module => 'Kernel::System::CustomerUser::LDAP',
    Params => {
        # ldap host
        Host => 'bay.csu Hayward.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 => "",
        # in case you want to add always one filter to each ldap query, use
        # this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter => '(objectclas
        AlwaysFilter => "",
        # if your frontend is e. g. iso-8859-1 and the charset of your
        # ldap server is utf-8, use this options (if not, ignore it)
#         SourceCharset => 'utf-8',
#         DestCharset => 'iso-8859-1',
        # Net::LDAP new params (if needed - for more info see perldoc Net::LDAP)
        Params => {
            port => 389,
            timeout => 120,
            async => 0,
            version => 3,
        },
    },
},

```

```

# customer uniq id
CustomerKey => 'uid',
# customer #
CustomerID => 'mail',
CustomerUserListFields => ['cn', 'mail'],
CustomerUserSearchFields => ['uid', 'cn', 'mail'],
CustomerUserSearchPrefix => "",
CustomerUserSearchSuffix => '*',
CustomerUserSearchListLimit => 250,
CustomerUserPostMasterSearchFields => ['mail'],
CustomerUserNameFields => ['givenname', 'sn'],
# show now own tickets in customer panel, CompanyTickets
CustomerUserExcludePrimaryCustomerID => 0,
# add a ldap filter for valid users (expert setting)
#
CustomerUserValidFilter => '(! (description=gesperrrt))',
# admin can't change customer preferences
AdminSetPreferences => 0,
Map => [
  # note: Login, Email and CustomerID needed!
  # var, frontend, storage, shown (1=always,2=lite), required, storage-type, http
  [ 'UserSalutation', 'Title', 'title', 1, 0, 'var', "", 0 ],
  [ 'UserFirstname', 'Firstname', 'givenname', 1, 1, 'var', "", 0 ],
  [ 'UserLastname', 'Lastname', 'sn', 1, 1, 'var', "", 0 ],
  [ 'UserLogin', 'Username', 'uid', 1, 1, 'var', "", 0 ],
  [ 'UserEmail', 'Email', 'mail', 1, 1, 'var', "", 0 ],
  [ 'UserCustomerID', 'CustomerID', 'mail', 0, 1, 'var', "", 0 ],
#
  [ 'UserCustomerIDs', 'CustomerIDs', 'second_customer_ids', 1, 0, 'var', "", 0 ],
  [ 'UserPhone', 'Phone', 'telephonenumber', 1, 0, 'var', "", 0 ],
  [ 'UserAddress', 'Address', 'postaladdress', 1, 0, 'var', "", 0 ],
  [ 'UserComment', 'Comment', 'description', 1, 0, 'var', "", 0 ],
],
};

```

It is possible to integrate up to 10 different customer user backends. With the customer user management interface in OTRS all customer data can be viewed or edited (if write access is possible).

9.3. Backends to authenticate agents and customer users

OTRS offers the possibility to authenticate agents and customers against different backends.

9.3.1. Authentication backends for agents

9.3.1.1. DB (Default)

The backend to authenticate agents which is used per default by OTRS is the OTRS database. Agents can be added and edited via the user management interface in the admin area

Example 9-6. Authenticate agents against a DB backend

```
$Self->{'AuthModule'} = 'Kernel::System::Auth::DB';
```

9.3.1.2. LDAP

If a LDAP directory has all your agent data stored you can use the LDAP module to authenticate your users in OTRS. This module has only read access to the LDAP tree, that means you can't edit your users via the user management interface

Example 9-7. Authenticate agents against a LDAP backend

```
# 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';
# for ldap posixGroups objectclass (just uid)
# $Self->{'AuthModule::LDAP::UserAttr'} = 'UID';
# for non ldap posixGroups objectclass (with full user dn)
# $Self->{'AuthModule::LDAP::UserAttr'} = 'DN';

# 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'} = "";

# in case you want to add always one filter to each ldap query, use
# this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter => '(objectclass=user)'
$Self->{'AuthModule::LDAP::AlwaysFilter'} = "";
```

```

# in case you want to add a suffix to each login name, then
# you can use this option. e. g. user just want to use user but
# in your ldap directory exists user@domain.
#   $Self->{'AuthModule::LDAP::UserSuffix'} = '@domain.com';

# Net::LDAP new params (if needed - for more info see perldoc Net::LDAP)
$Self->{'AuthModule::LDAP::Params'} = {
    port => 389,
    timeout => 120,
    async => 0,
    version => 3,
};

```

The following configuration parameters can be used to synchronize the user data from your LDAP directory into your local OTRS database. This reduces the requests to your LDAP server and speeds up the authentication with OTRS. The data sync is done when the agent authenticates the first time. Although the data can be synched into the local OTRS database the LDAP directory is the last instance for the authentication, so a inactive user in the LDAP tree can't authenticate to OTRS even when the account data are allready stored in the OTRS database. The agent data in the LDAP directory can't be edited via the web interface of OTRS, because only read access is given, so the data have to be managed directly in the LDAP tree.

```

# UserSyncLDAPMap
# (map if agent should create/synced from LDAP to DB after login)
$Self->{UserSyncLDAPMap} = {
    # DB -> LDAP
    Firstname => 'givenName',
    Lastname => 'sn',
    Email => 'mail',
};

# UserSyncLDAPGroups
# (If "LDAP" was selected for AuthModule, you can specify
# initial user groups for first login.)
$Self->{UserSyncLDAPGroups} = [
    'users',
];

# UserTable
$Self->{DatabaseUserTable} = 'system_user';
$Self->{DatabaseUserTableUserID} = 'id';
$Self->{DatabaseUserTableUserPW} = 'pw';
$Self->{DatabaseUserTableUser} = 'login';

```

9.3.1.3. HTTPBasicAuth for agents

If you want to implement a "single sign on" solution for all your agents, you can use http basic authentication (for all your systems) and the HTTPBasicAuth module for OTRS (the OTRS login is not needed any more).

Example 9-8. Authenticate agents via HTTPBasic

```
# This is an example configuration for an apache ($ENV{REMOTE_USER})
# auth. backend. Use it if you want to have a single login through
# apache http-basic-auth
$Self->{'AuthModule'} = 'Kernel::System::Auth::HTTPBasicAuth';

# Note:
#
# If you use this module, you should use as fallback
# the following config settings if user isn't login through
# apache ($ENV{REMOTE_USER})
$Self->{'LoginURL'} = 'http://host.example.com/not-authorized-for-otrs.html';
$Self->{'LogoutURL'} = 'http://host.example.com/thanks-for-using-otrs.html';
```

9.3.1.4. Radius

The following configuration parameters can be used to authenticate agents against a radius server.

Example 9-9. Authenticate agents against a radius backend

```
# This is example configuration to auth. agents against a radius server
$Self->{'AuthModule'} = 'Kernel::System::Auth::Radius';
$Self->{'AuthModule::Radius::Host'} = 'radiushost';
$Self->{'AuthModule::Radius::Password'} = 'radiussecret';
```

9.3.2. Authentication backends for customer users

9.3.2.1. Database (Default)

The authentication backend for customer users which is used per default by OTRS is the OTRS database. With this backend all customer data can be edited via the web interface of OTRS.

Example 9-10. Customer user authentication against a DB backend

```
# This is the auth. module against the otrs db
$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::DB';
$Self->{'Customer::AuthModule::DB::Table'} = 'customer_user';
$Self->{'Customer::AuthModule::DB::CustomerKey'} = 'login';
$Self->{'Customer::AuthModule::DB::CustomerPassword'} = 'pw';
# $Self->{'Customer::AuthModule::DB::DSN'} = "DBI:mysql:database=customerdb;host=customerdb";
# $Self->{'Customer::AuthModule::DB::User'} = "some_user";
# $Self->{'Customer::AuthModule::DB::Password'} = "some_password";
```

9.3.2.2. LDAP

If you have a LDAP directory with all your customer data you can use the LDAP module to authenticate your customer users to OTRS. Because this module has only read access to the LDAP backend, it is not possible to edit the customer data via the web interface of OTRS.

Example 9-11. Customer user authentication against a LDAP backend

```
# 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';
# for ldap posixGroups objectclass (just uid)
$Self->{'Customer::AuthModule::LDAP::UserAttr'} = 'UID';
# for non ldap posixGroups objectclass (full user dn)
# $Self->{'Customer::AuthModule::LDAP::UserAttr'} = 'DN';

# 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'} = "";

# in case you want to add always one filter to each ldap query, use
# this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter => '(objectclass=user)'
$Self->{'Customer::AuthModule::LDAP::AlwaysFilter'} = "";

# in case you want to add a suffix to each customer login name, then
# you can use this option. e. g. user just want to use user but
```

```
# in your ldap directory exists user@domain.
#   $Self->{'Customer::AuthModule::LDAP::UserSuffix'} = '@domain.com';

# Net::LDAP new params (if needed - for more info see perldoc Net::LDAP)
$Self->{'Customer::AuthModule::LDAP::Params'} = {
    port => 389,
    timeout => 120,
    async => 0,
    version => 3,
};
```

9.3.2.3. HTTPBasicAuth for customer users

If you want to implement a "single sign on" solution for all your customer users, you can use HTTPBasic authentication (for all your systems) and use the HTTPBasicAuth module with OTRS (no login is needed with OTRS any more).

Example 9-12. Customer user authentication with HTTPBasic

```
# This is an example configuration for an apache ($ENV{REMOTE_USER})
# auth. backend. Use it if you want to have a single login through
# apache http-basic-auth
$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::HTTPBasicAuth';

# Note:
# If you use this module, you should use the following
# config settings as fallback, if user isn't login through
# apache ($ENV{REMOTE_USER})
$Self->{'CustomerPanelLoginURL'} = 'http://host.example.com/not-authorized-for-otrs.html';
$Self->{'CustomerPanelLogoutURL'} = 'http://host.example.com/thanks-for-using-otrs.html';
```

9.3.2.4. Radius

The following settings can be used to authenticate your customer users against a radius server.

Example 9-13. Customer user authentication against a radius backend

```
# This is a example configuration to auth. customer against a radius server
$Self->{'Customer::AuthModule'} = 'Kernel::System::Auth::Radius';
$Self->{'Customer::AuthModule::Radius::Host'} = 'radiushost';
$Self->{'Customer::AuthModule::Radius::Password'} = 'radiussecret';
```

9.4. Customize the customer self registration

It is possible to customize the self registration for new customer users which is reachable via the customer.pl panel. New optional or required fields like phone number, address or state can be added.

The following example shows how a required field for the phone number can be specified.

9.4.1. Customizing the web interface

To display the new field for the phone number in the customer.pl web interface the .dtl file which is responsible for the layout in this interface has to be modified. Edit the Kernel/Output/HTML/Standard/CustomerLogin.dtl file and add the new field around line 128.

```
[...]
<tr>
  <td>$Text{"Phonenumber"}: </td>
  <td><input type="text" name="Phone" value="$QData{"UserPhone"}" size="20" maxlength="
</tr>
[...]
```

9.4.2. Customer mapping

In the next step the customer mapping has to be expanded with the new entry for the phone number. To ensure that the changes are not lost after an update, put the "CustomerUser" settings from the Kernel/Config/Defaults.pm into the Kernel/Config.pm. Now change the MAP array and add the new phone number field:

```
# CustomerUser
# (customer user database backend and settings)
$Self->{CustomerUser} = {
  Name => 'Database Backend',
  Module => 'Kernel::System::CustomerUser::DB',
  Params => {
    # if you want to use an external database, add the
    # required settings
```



```

#         DSN => 'DBI:odbc:yourdsn',
#         DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',
#         User => "",
#         Password => "",
#         Table => 'customer_user',
    },
    # customer uniq id
    CustomerKey => 'login',
    # customer #
    CustomerID => 'customer_id',
    CustomerValid => 'valid_id',
    CustomerUserListFields => ['first_name', 'last_name', 'email'],
#     CustomerUserListFields => ['login', 'first_name', 'last_name', 'customer_id', 'ema
    CustomerUserSearchFields => ['login', 'last_name', 'customer_id'],
    CustomerUserSearchPrefix => "",
    CustomerUserSearchSuffix => '*',
    CustomerUserSearchListLimit => 250,
    CustomerUserPostMasterSearchFields => ['email'],
    CustomerUserNameFields => ['salutation', 'first_name', 'last_name'],
    CustomerUserEmailUniqCheck => 1,
#     # show now own tickets in customer panel, CompanyTickets
#     CustomerUserExcludePrimaryCustomerID => 0,
#     # generate auto logins
#     AutoLoginCreation => 0,
#     AutoLoginCreationPrefix => 'auto',
#     # admin can change customer preferences
#     AdminSetPreferences => 1,
#     # just a read only source
#     ReadOnly => 1,
    Map => [
        # note: Login, Email and CustomerID needed!
        # var, frontend, storage, shown (1=always,2=lite), required, storage-type, http
        [ 'UserSalutation', 'Salutation', 'salutation', 1, 0, 'var', "", 0 ],
        [ 'UserFirstname', 'Firstname', 'first_name', 1, 1, 'var', "", 0 ],
        [ 'UserLastname', 'Lastname', 'last_name', 1, 1, 'var', "", 0 ],
        [ 'UserLogin', 'Username', 'login', 1, 1, 'var', "", 0 ],
        [ 'UserPassword', 'Password', 'pw', 0, 1, 'var', "", 0 ],
        [ 'UserEmail', 'Email', 'email', 0, 1, 'var', "", 0 ],
#         [ 'UserEmail', 'Email', 'email', 1, 1, 'var', '$Env{"CGIHandle"}' ],
        [ 'UserCustomerID', 'CustomerID', 'customer_id', 0, 1, 'var', "", 0 ],
#         [ 'UserCustomerIDs', 'CustomerIDs', 'customer_ids', 1, 0, 'var', "", 0 ],
        [ 'UserComment', 'Comment', 'comments', 1, 0, 'var', "", 0 ],
        [ 'UserPhone', 'Phone', 'phone', 1, 0, 'var', "", 0 ],
        [ 'ValidID', 'Valid', 'valid_id', 0, 1, 'int', "", 0 ],
    ],
    # default selections
    Selections => {
        UserSalutation => {
            'Mr.' => 'Mr.',
            'Mrs.' => 'Mrs.',
        },
    },
};

```

9.4.3. Customize the customer_user table in the OTRS DB

The last step is to add the new phone number column to the customer_user table in the OTRS database. In this column the entries for the phone numbers will be stored.

```
linux:~# mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6 to server version: 5.0.18-Debian_7-log

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

mysql> use otrs;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> ALTER TABLE customer_user ADD phone VARCHAR (200);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> quit
Bye
linux:~#
```

All needed settings were done and the new field for the phone should be displayed in the customer.pl panel. New customer users should have to insert their phone number if they register a new account. If your apache is configured to use mod_perl for OTRS, you should restart the web server to activate the changes.

Chapter 10. Modifying ticket states and ticket state types

It is possible to change the predefined states and state types for tickets in OTRS. Also new states or state types can be added and used with the ticket system. Two options are important for a state, the name (state-name) and the type of the state (state-type).

- The default states of OTRS are 'closed successful', 'closed unsuccessful', 'merged', 'new', 'open', 'pending auto close+', 'pending auto close-', 'pending reminder' and 'removed'.
- Every state must have a name (status-name) and a status type (status-type) has to be linked to this state. The name of a state is free selectable and can be specified via the admin area of OTRS in the section for states. The state types can not be changed via the web interface, the database has to be modified if new types are needed or if the name of types have to be changed.

In the section for states in the admin area of OTRS the name of already existing states can be changed or new states can be added. If the state "new" has been changed via the web interface, this change also has to be configured via the config file `Kernel/Config.pm` or via the SysConfig interface. The following settings have to be modified to ensure that OTRS works with the changed state for "new":

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

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

If the state "open" has been changed, the configuration of OTRS also has to be modified to ensure that OTRS uses the new settings:

```
[...]
# default phone new state
$Self->{'Ticket::Frontend::PhoneNextState'} = 'open';

# PostmasterFollowUpState
# (The state if a ticket got a follow up.) [default: open]
```

```
$Self->{PostmasterFollowUpState} = 'open';  
[...]
```

If a new state type should be added the ticket_state_type table in the OTRS database needs to be modified with a database client program.

```
linux:~# mysql -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 23 to server version: 5.0.16-Debian_1-log  
  
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.  
  
mysql> use otrs;  
Reading table information for completion of table and column names  
You can turn off this feature to get a quicker startup with -A  
  
Database changed  
mysql> insert into ticket_state_type (name,comments) values ('own','Own  
state type');  
Query OK, 1 row affected (0.00 sec)  
  
mysql> quit  
Bye  
linux:~#
```

Now it is possible to use the new state type "own" via the web interface in the section for states in the admin area. After a state has been linked with this new state type also the configuration of OTRS has to be changed to ensure that the new state is used. Just modify your configuration files or use the SysConfig interface to change the following settings:

```
[...]  
# Ticket::DefaultNextMoveStateType  
# default move next state  
$Self->{'Ticket::DefaultNextMoveStateType'} = ['open', 'closed'];  
  
# next possible states after phone  
$Self->{'Ticket::PhoneDefaultNextStateType'} = ['open', 'pending auto', 'pending remind'];  
  
# default next state  
$Self->{'Ticket::Frontend::PhoneNextState'} = 'closed successful';  
  
# default next state [default: open]  
$Self->{'Ticket::Frontend::PhoneNewNextState'} = 'open';
```

```
# next possible states after email
$Self->{'Ticket::EmailDefaultNextStateType'} = ['own-state', 'open', 'pending auto', 'p

# default next state
$Self->{'Ticket::Frontend::EmailNewNextState'} = 'open';

# (default note next state)
$Self->{'Ticket::DefaultNextNoteStateType'} = ['new', 'open', 'closed'];

# Ticket::DefaultNextOwnerStateType
# (default note next state)
$Self->{'Ticket::DefaultNextOwnerStateType'} = ['open', 'closed'];

# default compose next state
$Self->{'Ticket::DefaultNextComposeType'} = 'open';

# next possible states for compose message
$Self->{'Ticket::DefaultNextComposeStateType'} = ['open', 'closed', 'pending auto', 'pe

# default bounce next state
$Self->{'Ticket::Frontend::BounceState'} = 'closed successful';

# next possible states for bounce message
$Self->{'Ticket::DefaultNextBounceStateType'} = ['open', 'closed'];

# next possible states for forward message
$Self->{'Ticket::DefaultNextForwardStateType'} = ['open', 'closed'];

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

# Ticket::UnlockStateType
# (Tickets which can be unlocked by bin/UnlockTickets.pl
# (see http://yourhost/otrs/index.pl?Action=AdminState -> StateType)
$Self->{'Ticket::UnlockStateType'} = ['open', 'new'];
[...]
```

Just add your new state to the config settings where the new state should be used.

Chapter 11. Modifying ticket priorities

Please perform the following steps to change / modify the ticket priorities in OTRS. The changes can't be done via the web interface, the OTRS database has to be changed directly.

- Use a database client to connect to your database server and select the OTRS database. MySQL is used in the following example:

```
linux:~# mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10 to server version: 5.0.18-Debian_4-log

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

mysql> USE otrs;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql>
```

- The following SQL statement shows 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 defines the order of the priorities, 1 is the minimum, 5 or above sets a higher priority. The number in the name of a priority is used by the system to ensure the correct order of the different prios.

- Modifying / changing a priority via SQL:

```
mysql> UPDATE ticket_priority SET name = '3 default' WHERE id = 3;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql>
```

This SQL statement changes the name of the priority "3 normal" to "3 default".

- If a new priority was added or if a priority was changed, this changes also have to be set in the config file of OTRS or via the SysConfig interface:

```
[...]
# PostmasterDefaultPriority
# (The default priority of new tickets.) [default: '3 normal']
$Self->{PostmasterDefaultPriority} = '3 default';
[...]
# Ticket::Frontend::EmailPriority
# default priority for email tickets [default: 3 normal]
$Self->{'Ticket::Frontend::EmailPriority'} = '3 default';
[...]
# default phone priority [default: 3 normal]
$Self->{'Ticket::Frontend::PhonePriority'} = '3 default';
[...]
# CustomerDefaultPriority
# (default priority of new customer tickets)
$Self->{CustomerDefaultPriority} = '3 default';
[...]
```

If you like to add a new priority, update the `ticket_priority` table in the OTRS database. Take care, that the ID and the number in the priority name reflects the urgency of the new priority.

Chapter 12. Creating own themes

You can create your own themes for OTRS to use the layout you like in the web frontend. To create own themes the output template have to be customized to your needs.

More information about the syntax and structure of output templates can be found in the Developer Manual on <http://doc.otrs.org> (<http://doc.otrs.org>) , especialy in the chapter about *templates* (<http://doc.otrs.org/developer/2.0/en/html/c494.html>) .

To create a new theme called "Company" perform the following steps:

1. Copy the directory `Kernel/Output/HTML/Standard` to `Kernel/Output/HTML/Company`.
2. Customize the files in the directory `Kernel/Output/HTML/Company` and change the layout to you needs.
3. To activate the new theme for OTRS the database has to be changed manually and the new theme has to be added to the theme table in the OTRS database. Use the following commands with MySQL:

```
linux:~# mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 26 to server version: 5.0.22-Debian_2-log

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

mysql> use otrs;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> INSERT INTO theme
->      (theme, valid_id, create_time, create_by, change_time, change_by)
->      VALUES
->      ('Company', 1, current_timestamp, 1, current_timestamp, 1);
mysql>
```

Now the new theme should be useable in OTRS. It should be possible to select it via your personal preferences page.

Chapter 13. Localization of the OTRS frontend

The frontend of OTRS supports different languages. The `Kernel/Language/*.pm` files are the different files for localization.

How localization for the OTRS framework is done and what steps you have to perform to create a translation for a new language or how you can setup custom translations can be found in the "Language Translations" (<http://doc.otrs.org/developer/2.0/en/html/c630.html>) chapter from the developer manual on <http://doc.otrs.org> (<http://doc.otrs.org>) .

Chapter 14. PGP

OTRS is able to sign or encrypt outgoing messages with PGP. Also encrypted incoming messages can be decrypted. Encryption and decryption is done with the GPL tool GnuPG. To setup GnuPG for OTRS the following steps have to be performed:

1. The first step is to install GnuPG via the package manager of your operating system.
2. In the next step GnuPG has to be configured for the usage with OTRS. The needed directories for GnuPG and a private key have to be created. The following command has to be executed as OTRS user from a shell.

```
linux:~# su otrs
linux:/root$ cd
linux:~$ pwd
/opt/otrs
linux:~$ gpg --gen-key
gpg (GnuPG) 1.4.2; Copyright (C) 2005 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

gpg: directory '/opt/otrs/.gnupg' created
gpg: new configuration file '/opt/otrs/.gnupg/gpg.conf' created
gpg: WARNING: options in '/opt/otrs/.gnupg/gpg.conf' are not yet active during t
his run
gpg: keyring '/opt/otrs/.gnupg/secring.gpg' created
gpg: keyring '/opt/otrs/.gnupg/pubring.gpg' created
Please select what kind of key you want:
  (1) DSA and Elgamal (default)
  (2) DSA (sign only)
  (5) RSA (sign only)
Your selection? 1
DSA keypair will have 1024 bits.
ELG-E keys may be between 1024 and 4096 bits long.
What keysize do you want? (2048)
Requested keysize is 2048 bits
Please specify how long the key should be valid.
  0 = key does not expire
  <n> = key expires in n days
  <n>w = key expires in n weeks
  <n>m = key expires in n months
  <n>y = key expires in n years
Key is valid for? (0)
Key does not expire at all
Is this correct? (y/N) y
```

You need a user ID to identify your key; the software constructs the user ID from the Real Name, Comment and Email Address in this form:

```
gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m, 0f, 1u
pub   1024D/7245A970 2006-02-03
      Key fingerprint = 2ED5 BC36 D2B6 B055 7EE1  5833 1D7B F967 7245 A970
uid           Ticket System (Private gpg key for ticket system with address support@example.com) <support@example.com>
sub   2048g/52B97069 2006-02-03

linux:~$
```

3. Now OTRS has to be prepared to use PGP. Open the SysConfig interface via the admin panel and search for "PGP". Select the sub group Crypt::PGP from the search results.

In the screen for the PGP settings PGP should be activated for OTRS (first option). Also the path to the gpg program should be checked.

The next config setting (PGP::Options) might to be changed. Via this config setting the parameters that are used for every execution of gpg by the OTRS user can be specified. Especially the directory of the config files for GnuPG of the OTRS user is important. In the example `/opt/otrs/.gnupg` is used. This directory was created in the first step of the PGP configuration.

Via the last config option it is possible to specify the pairs of key IDs and their passwords for the own private keys. Because communication partners from outside write to the ticket system and their messages are encrypted with the public key, OTRS can decrypt these messages with the ID/passwords specified here.

How to get the id of your own private key? The ID of your own private key is already shown during the key generation (see step 1 from above). It is also possible to get the ID if the following command is executed as OTRS user:

```
linux:~# su otrs
linux:/root$ cd
linux:~$ pwd
/opt/otrs
linux:~$ gpg --list-keys
/opt/otrs/.gnupg/pubring.gpg
-----
pub 1024D/7245A970 2006-02-03
uid Ticket System (Private pgp key for ticket system with
address support@example.com) <support@example.com>
sub 2048g/52B97069 2006-02-03

linux:~$
```

The ID of the private key can be found in the line that starts with "sub". It is a hexa decimal string that is eight characters long, in the example above it is "52B97069". The password you have to specify for this key in the ticket system is the same like given during key generation.

After these data have been inserted the "Update" button can be used to store the settings. OTRS is ready to receive and decrypt encoded messages now.

4. The last step is the import of a customers public key. This ensures that also encrypted messages can be sent out to this customer. There are two possibilities to import a public key of a customer.

The first possibility is to specify the public key of a customer in the interface for the customer management. A public can be given if a customer is created or modified.

The second possibility is to specify the key via the PGP settings reachable from the admin area of OTRS. On the right side of this screen all already imported public keys of customers are displayed. After PGP has been activated and configured for OTRS your own public key should be listed there also. In the left area of the PGP setting screen it is possible to search for keys. Also a new public key can be uploaded into the system from a file.

The files with the public key that need to be imported into OTRS have to be GnuPG conform key files. In most cases the key stored in a file is an “ASCII armored key”.. OTRS can deal with this format.

Chapter 15. S/MIME

At first sight, encryption with S/MIME seems to be a little bit more complicated than encryption with PGP. First of all, you have to establish a Certification Authority (CA) for the OTRS system. Then, the following procedures are very much like those you have to perform using PGP: configure OTRS, install your own certificate, import other public certificates if needed etc.

The S/MIME configuration is conducted outside the OTRS web interface for the most part, and should be carried out in a shell by the OTRS user, i.e. with his/her user rights. The MIME configuration under Linux is based on SSL (OpenSSL). Therefore, check first of all whether the OpenSSL package is installed on your system. The OpenSSL package includes a script called CA.pl with which the most important steps of certificate creation can be performed. To simplify the procedure, find out where in the file system the CA.pl script is stored and enter the location temporarily into the path variable of the shell.

```
otrs@linux:~> rpm -ql openssl | grep CA
/usr/share/ssl/misc/CA.pl
otrs@linux:~> export PATH=$PATH:/usr/share/ssl/misc
otrs@linux:~> which CA.pl
/usr/share/ssl/misc/CA.pl
otrs@linux:~> mkdir tmp; cd tmp
otrs@linux:~/tmp>
```

The example shows that a new temporary directory ~/tmp has been created in which the certificate is generated.

To create a certificate, perform the following operations in the command line (we assume that the OTRS administrator has to create a SSL certificate for test and learning purposes. In case you already have a certified SSL certificate for the encryption, use it and skip the following):

1. Establish your own Certification Authority for SSL. You need it to certify the request for your own SSL certificate.

```
otrs@linux:~/tmp> CA.pl -newca
CA certificate filename (or enter to create)

Making CA certificate ...
Generating a 1024 bit RSA private key
...+++++
.....+++++
writing new private key to './demoCA/private/cakey.pem'
Enter PEM pass phrase:
```

```

Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:OTRS-state
Locality Name (eg, city) []:OTRS-town
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Your company
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:OTRS Admin
Email Address []:otrs@your-domain.tld
otrs@linux:~/tmp> ls -la demoCA/
total 8
-rw-r--r--  1 otrs otrs 1330 2006-01-08 17:54 cacert.pem
drwxr-xr-x  2 otrs otrs  48 2006-01-08 17:53 certs
drwxr-xr-x  2 otrs otrs  48 2006-01-08 17:53 crl
-rw-r--r--  1 otrs otrs   0 2006-01-08 17:53 index.txt
drwxr-xr-x  2 otrs otrs  48 2006-01-08 17:53 newcerts
drwxr-xr-x  2 otrs otrs  80 2006-01-08 17:54 private
-rw-r--r--  1 otrs otrs  17 2006-01-08 17:54 serial
otrs@linux:~/tmp>

```

2. Generate a certificate request.

```

otrs@linux:~/tmp> CA.pl -newreq
Generating a 1024 bit RSA private key
.....++++++
....++++++
writing new private key to 'newreq.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:DE\keyreturn
State or Province Name (full name) [Some-State]:OTRS-state
Locality Name (eg, city) []:OTRS-town
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Your company
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:OTRS admin

```

```
Email Address []:otrs@your-domain.tld

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Request (and private key) is in newreq.pem
otrs@linux:~/tmp> ls -la
total 4
drwxr-xr-x  6 otrs otrs  232 2006-01-08 17:54 demoCA
-rw-r--r--  1 otrs otrs 1708 2006-01-08 18:04 newreq.pem
otrs@linux:~/tmp>
```

3. Signing of the certificate request. The certificate request can either be signed, and thereby certified, by your own CA or, which is more respectable, by another external certified CA.

```
otrs@linux:~/tmp> CA.pl -signreq
Using configuration from /etc/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number:
        fd:85:f6:9f:14:07:16:c8
    Validity
        Not Before: Jan  8 17:04:37 2006 GMT
        Not After : Jan  8 17:04:37 2007 GMT
    Subject:
        countryName             = DE
        stateOrProvinceName     = OTRS-state
        localityName            = OTRS-town
        organizationName        = Your Company
        commonName              = OTRS administrator
        emailAddress            = otrs@your-domain.tld
    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        Netscape Comment:
            OpenSSL Generated Certificate
        X509v3 Subject Key Identifier:
            01:D9:1E:58:C0:6D:BF:27:ED:37:34:14:D6:04:AC:C4:64:98:7A:22
        X509v3 Authority Key Identifier:
            keyid:10:4D:8D:4C:93:FD:2C:AA:9A:B3:26:80:6B:F5:D5:31:E2:8E:DB:A8
            DirName:/C=DE/ST=OTRS-state/L=OTRS-town/O=Your Company/
            CN=OTRS admin/emailAddress=otrs@your-domain.tld
            serial:FD:85:F6:9F:14:07:16:C7

Certificate is to be certified until Jan  8 17:04:37 2007 GMT (365 days)
Sign the certificate? [y/n]:y
```



```

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Signed certificate is in newcert.pem
otrs@linux:~/tmp>

```

4. Generate your own certificate, and all data going with it, using the signed certificate request.

```

otrs@linux:~/tmp> CA.pl -pkcs12 "OTRS Certificate"
Enter pass phrase for newreq.pem:
Enter Export Password:
Verifying - Enter Export Password:
otrs@linux:~/tmp> ls -la
total 12
drwxr-xr-x  6 otrs otrs  328 2006-01-08 18:04 demoCA
-rw-r--r--  1 otrs otrs 3090 2006-01-08 18:13 newcert.p12
-rw-r--r--  1 otrs otrs 3791 2006-01-08 18:04 newcert.pem
-rw-r--r--  1 otrs otrs 1708 2006-01-08 18:04 newreq.pem
otrs@linux:~/tmp>

```

these operations have been performed, the S/MIME setup must be completed in OTRS.

This part of the setup is carried out in the admin area, block system, choosing the menu item "SMIME". In case the general S/MIME support in OTRS has not yet been enabled, the mask points this out to the administrator and provides a comfortable link for that purpose.

With the SysConfig, you can enable and configure the general S/MIME support. You can find this option in the SysConfig group "Crypt::SMIME".

Here you can activate the S/MIME support, define the paths for the OpenSSL command and the directory for the certificates. The key file created above must be stored in the directory indicated here. Otherwise OpenSSL cannot use it.

The next step is performed in the S/MIME configuration in the OTRS admin area (again). Here, you can import the private key(s) of the OTRS system and the public keys of other communication partners. Enter the public key that has been created in the beginning of this section and has been added in OTRS.

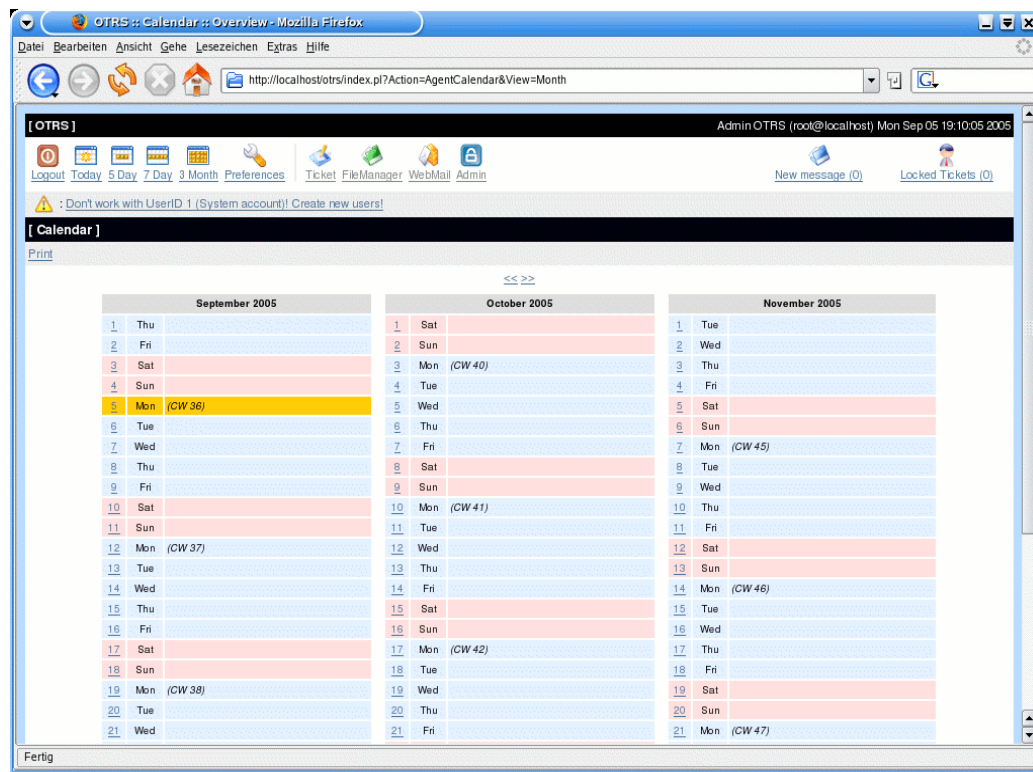
Obviously, all public S/MIME keys of communication partners can be imported using the customer administration tool as well, when creating a new entry or modifying an existing one.

Chapter 16. Additional applications

With OTRS 2.0 and subsequent versions it is possible to install additional applications that can be used within the central framework. The installation can be done with the package manager via the adminarea. The package manager downloads the applications from an online repository and manages package dependencies.

16.1. Calendar

You can edit and view both private and public user-based appointments With the web-based calendar.

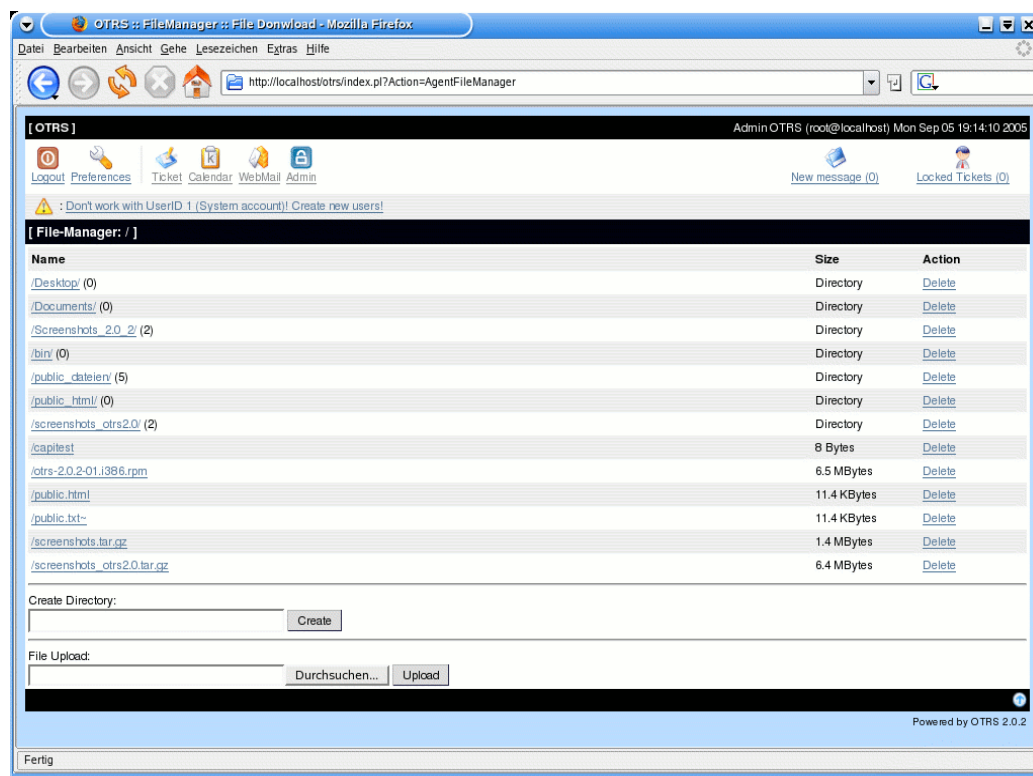


16.2. Content manager

The content manager lets you edit and administrate your web projects online and via your web browser.

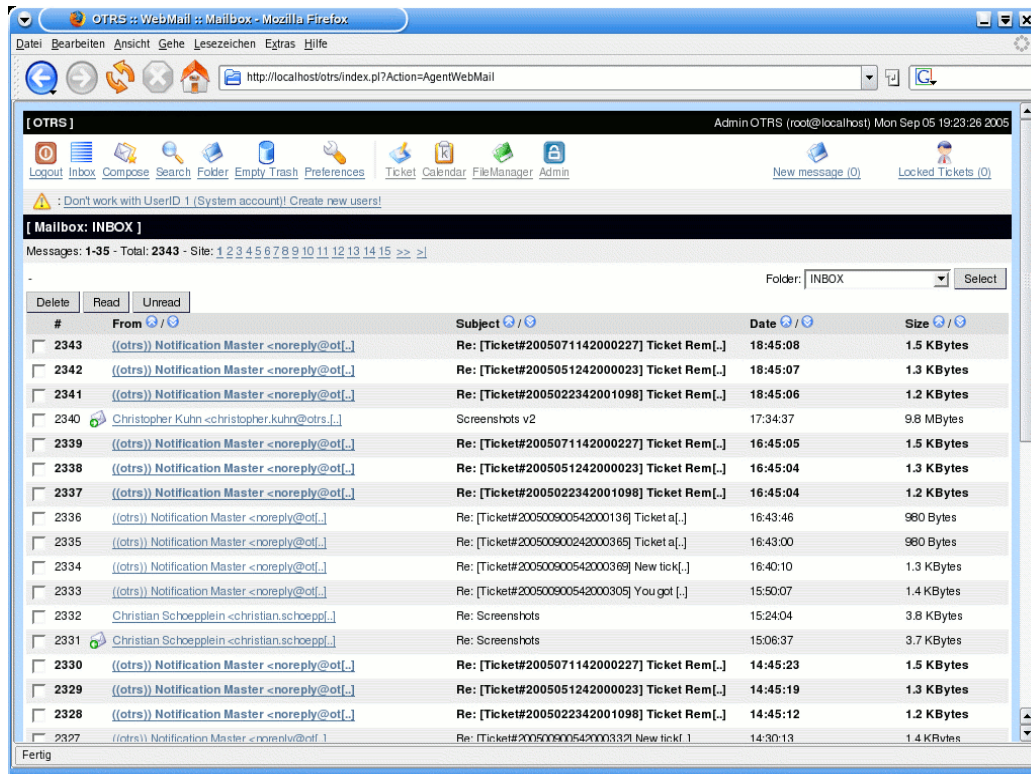
16.3. File manager

The file manager makes it possible to access the content of a directory that is located in the file system of the machine where OTRS is installed. You can use your web browser to upload, download, delete and view files, browse and create subdirectories, etc.



16.4. Web mailer

With the webmailer you can access your mailbox via an IMAP server. It is possible to compose new messages, answer and forward your mails, etc. simply with your web browser.



16.5. FAQ

The FAQ system is at present still integrated in the central OTRS framework. It supports editing and viewing FAQ articles. To improve browsing of the different articles, they can be structured in categories and sorted by language.

16.6. System status

With the system status application the OTRS admin can give an overview of different system tasks. The output of this module is reachable through the adminarea of OTRS.

OTRS :: Admin :: Status - Mozilla Firefox

http://localhost/otrs/index.pl?Action=AdminSystemStatus

[Free memory in MBytes (free -m)]

	total	used	free	shared	buffers	cached
Mem:	503	496	6	0	7	169
-/+ buffers/cache:		319	183			
Swap:	1027	113	913			

[Report process status (ps -ax | grep httpd)]

PID	PPID	STATUS	USER	COMMAND
8761	?	Ss		0:00 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf
8762	?	S		0:17 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf
8763	?	S		0:17 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf
8764	?	S		0:12 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf
8765	?	S		0:34 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf
8766	?	S		0:12 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf
8826	?	S		0:28 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf
8830	?	S		0:14 /usr/sbin/httpd2-prefork -f /etc/apache2/httpd.conf
11566	?	S		0:00 sh -c ps -ax grep httpd

[Show the 10 user, who logged in last (last -10)]

USER	TTY	FROM	LOGIN TIME	LOGOUT TIME	SESSION DURATION
tastatur	pts/2		Mon Sep 5 19:05	19:05	(00:00)
tastatur	pts/0		Mon Sep 5 19:05	still logged in	
tastatur	:0	console	Mon Sep 5 19:05	still logged in	
tastatur	:0		Mon Sep 5 19:05	19:05	(00:00)
tastatur	pts/1		Mon Sep 5 18:47	19:04	(00:17)
tastatur	pts/2		Mon Sep 5 18:44	18:44	(00:00)
tastatur	pts/0		Mon Sep 5 18:44	19:05	(00:20)
tastatur	:0	console	Mon Sep 5 18:44	19:04	(00:20)
tastatur	:0		Mon Sep 5 18:44	18:44	(00:00)
tastatur	pts/1		Mon Sep 5 12:59	15:12	(02:13)

wtmp begins Thu Sep 1 17:48:26 2005

[Show who is logged on and what they are doing (w)]

19:31:42 up 11:24, 3 users, load average: 0.44, 0.39, 0.34

USER	TTY	LOGIN@	IDLE	JCPU	PCPU	WHAT
tastatur	:0	19:05	?xdm?	5:51	0.10s	/bin/sh /usr/X11R6/bin/kde
tastatur	pts/0	19:05	26:23	0.00s	0.69s	kdcd [kdeinit] kdcd

Fertig

Chapter 17. 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.

17.1. OTRS

There are several options to improve the performance of OTRS.

17.1.1. 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 the ticket table. You will not have performance trouble until you have about 60,000 tickets (with 6,000 open tickets) in your system.
- Kernel::System::Ticket::IndexAccelerator::StaticDB, the most powerful module, it should be used when you have above 80,000 tickets (more than 6,000 open tickets) - use an extra ticket_index table, works like a view. Use bin/RebuildTicketIndex.pl for an initial index build.

17.1.2. 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 etc. in the database. Note: Don't use it for larger set ups.

Pro: If your webserver user isn't the otrs user, use this module to avoid file permission problems.

Contra: It's not really advisable 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 etc. in the local file system. Note: Use it for larger set ups.

Pro: Faster!

Contra: Your webserver user should be the otrs user (file system permissions!).

Note: If you use OTRS 1.2 or higher, you can switch from one to the other backend on the fly.

17.2. Database

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

17.2.1. MySQL

If you use MySQL table type MyISAM (which is the default) and have deleted a large part of a table or if you have made many changes to a table with variable-length rows (tables that have VARCHAR, BLOB or TEXT columns) you need to defragment the datafile (tables) with the "optimize" command.

You should try it if mysqld needs a lot of your CPU time. Optimize table ticket, ticket_history and article.

```
shell$ mysql -u user -p database
mysql$ optimize table ticket;
```

```
mysql$ optimize table ticket_history;
mysql$ optimize table article;
```

17.2.2. PostgreSQL

PostgreSQL is best tuned by modifying the postgresql.conf file in your PostgreSQL data directory. For advice on how to do this, see these articles:

<http://www.varlena.com/varlena/GeneralBits/Tidbits/perf.html>

(<http://www.varlena.com/varlena/GeneralBits/Tidbits/perf.html>)

http://www.varlena.com/varlena/GeneralBits/Tidbits/annotated_conf_e.html

(http://www.varlena.com/varlena/GeneralBits/Tidbits/annotated_conf_e.html) If performance is still not satisfactory, we suggest that you join the PostgreSQL Performance mailing list and ask questions there. The folks on the PostgreSQL list are very friendly and can probably help <http://www.postgresql.org/lists.html>.

17.3. Webserver

Of course you should use mod_perl 2.0 (<http://perl.apache.org/> (<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).

17.3.1. Preestablish database connections

You can have the database connections pre-established on process startup (httpd). This saves time (see README.webserver).

17.3.2. Preloaded modules - startup.pl

Use the scripts/apache-perl-startup.pl (mod_perl 1.0) or scripts/apache2-perl-startup.pl (mod_perl 2.0) startup script for preloaded/precompiled perl modules on your mod_perl webserver to be faster and smaller (see README.webserver).

17.3.3. Reload perl modules when updated on disk

Per default Apache::Reload (mod_perl 2.0) is used in scripts/apache2-httpd.include.conf. Disable it and you will get 8% more speed. But remember to restart the webserver if you change the Kernel/Config.pm

or any other modules! Important; now you can't use the OTRS-Packet-Manager via web interface, you need to use CMD - bin/opm.pl.

17.3.4. Choosing the Right Strategy

If you have a really large installation (over 1,000 new tickets a day and over 40 Agents) it is a good idea to read "Choosing the Right Strategy" (<http://perl.apache.org/docs/1.0/guide/strategy.html> (http://perl.apache.org/docs/1.0/guide/strategy.html)).

17.3.5. mod_gzip/mod_deflate

If your bandwidth is a little bit small, use mod_gzip for Apache1 (http://www.schroepl.net/projekte/mod_gzip/ (http://www.schroepl.net/projekte/mod_gzip/)) or mod_deflate for Apache2 (default module in Apache2). If you have an html page with 45k, mod_gzip/mod_deflate compresses it. The page will be about 7k.

17.3.6. mod_dosevasive

If you want to block http DoS (Denial of Service) attacks, use mod_dosevasive. (<http://www.nuclearelephant.com/projects/dosevasive/> (http://www.nuclearelephant.com/projects/dosevasive/)).

Chapter 18. Backing up the system

This chapter describes the backup and restore of the OTRS data.

18.1. Backup

There are two types of data to backup; the application (e.g. the files in `/opt/otrs` and the data stored in the database).

To simplify backups, the script `scripts/backup.pl` is included in every OTRS installation. It is able to backup all important data.

```
linux:/opt/otrs# cd scripts/
linux:/opt/otrs/scripts# ./backup.pl --help
backup.pl <Revision 1.1> - backup script
Copyright (c) 2001-2005 Martin Edenhofer <martin@otrs.org>
usage: backup.pl -d /data_backup/ [-c bzip2|gzip] [-r 30] [-t nofullbackup]
linux:/opt/otrs/scripts#
```

Execute the following command to create a backup:

```
linux:/opt/otrs/scripts# ./backup.pl -d /backup/
Backup /backup//2005-09-12_14-28/Config.tar.gz ... done
Backup /backup//2005-09-12_14-28/Application.tar.gz ... done
Dump MySQL rdbs ... done
Compress SQL-file... done
linux:/opt/otrs/scripts#
```

All data was stored in the directory `/backup/2005-09-12_14-28/`. additionally the data was saved into a `.tar.gz` file.

```
linux:/opt/otrs/scripts# ls /backup/2005-09-12_14-28/
Application.tar.gz Config.tar.gz DatabaseBackup.sql.gz
linux:/opt/otrs/scripts#
```

18.2. Restore

To restore a backup, the saved data for the application has to be written back into the installation directory, e.g. `/opt/otrs`. Also the database has to be restored.

A script `scripts/restore.pl` to simplify the restore process is shipped with every OTRS installation. It supports MySQL and PostgreSQL.

```
linux:/opt/otrs/scripts# ./restore.pl --help
restore.pl <Revision 1.1> - restore script
Copyright (c) 2001-2005 Martin Edenhofer <martin@otrs.org>
usage: restore.pl -b /data_backup/<TIME>/ -d /opt/otrs/
linux:/opt/otrs/scripts#
```

Data, which are stored in the directory `/backup/2005-09-12_14-28/`, can be restored with the following command and an OTRS installation in `/opt/otrs`.

```
linux:/opt/otrs/scripts# ./restore.pl -b /backup/2005-09-12_14-28 -d /opt/otrs/
Restore /backup/2005-09-12_14-28//Config.tar.gz ...
Restore /backup/2005-09-12_14-28//Application.tar.gz ...
create MySQL
decompresses SQL-file ...
cat SQL-file into MySQL database
compress SQL-file...
linux:/opt/otrs/scripts#
```

Appendix A. Additional Resources

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

A.1. Homepage OTRS.org

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



A.2. Mailinglists

The following table shows you our current community mailing lists.

Table A-1. Mailinglists

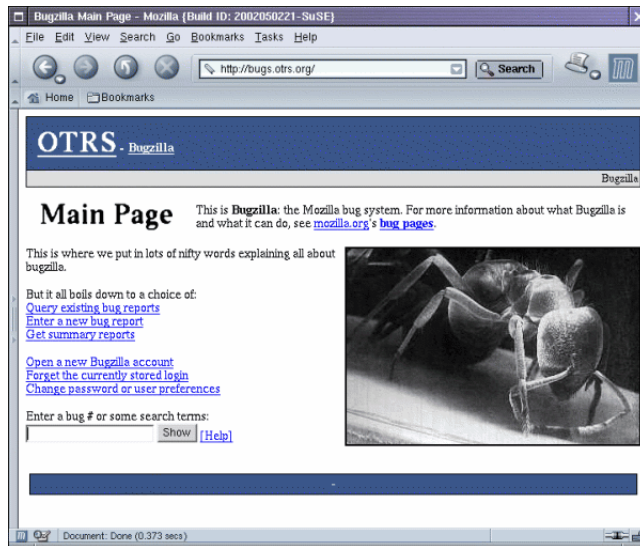
Name	Description	Homepage
announce at otrs.org	It is a low traffic list for announcements of new OTRS releases and security issues.	http://lists.otrs.org/cgi-bin/listinfo/announce (http://lists.otrs.org/cgi-bin/listinfo/announce)

Name	Description	Homepage
otrs at otrs.org	It is a medium to high traffic list with all sorts of questions on, and support for, the product.	http://lists.otrs.org/cgi-bin/listinfo/otrs (http://lists.otrs.org/cgi-bin/listinfo/otrs)
otrs-de at otrs.org	It is a medium to high traffic list with all sorts of questions on, and support for, the product. This list is in German.	http://lists.otrs.org/cgi-bin/listinfo/otrs-de (http://lists.otrs.org/cgi-bin/listinfo/otrs-de)
dev at otrs.org	It is a medium to high traffic list. The OTRS developers discuss various design and implementation issues here.	http://lists.otrs.org/cgi-bin/listinfo/dev (http://lists.otrs.org/cgi-bin/listinfo/dev)
doc-de at otrs.org	It is a low traffic list with all sorts of documentation questions, in German, on the product.	http://lists.otrs.org/cgi-bin/listinfo/doc-de (http://lists.otrs.org/cgi-bin/listinfo/doc-de)
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.	http://lists.otrs.org/cgi-bin/listinfo/i18n (http://lists.otrs.org/cgi-bin/listinfo/i18n)
cvs-log at otrs.org	It is a very high traffic list of CVS commits notifications.	http://lists.otrs.org/cgi-bin/listinfo/cvs-log (http://lists.otrs.org/cgi-bin/listinfo/cvs-log)

To subscribe to any of this lists visit the following link: <http://lists.otrs.org/> (<http://lists.otrs.org/>).

A.3. Bugtracking

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



You help us very much if you report bugs. We appreciate your help!

A.4. Commercial Support

For business assistance (support, consulting and training) you can contact the commercial part of OTRS, OTRS GmbH.

OTRS GmbH, Europaring 4, 94315 Straubing (Germany), Web: <http://otrs.com/> (<http://otrs.com/>)

Appendix B. Config Referenzliste

B.1. Framework

B.1.1. Core

B.1.1.1. SecureMode

Description	Value
Description:	If enabled, the webinstaller (installer.pl) can't be used.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SecureMode'} = '0';</code>

B.1.1.2. ProductName

Description	Value
Description:	The name of the application that is shown in the frontend.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'ProductName'} = 'OTRS';</code>

B.1.1.3. SystemID

Description	Value
Description:	The system identifier. Every ticket number and ID of a http session starts with this number.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SystemID'} = '10';</code>

B.1.1.4. FQDN

Description	Value
Description:	Full qualified domain name of your system.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'FQDN'} = 'yourhost.example.com';</code>

B.1.1.5. HttpType

Description	Value
Description:	If you want to use https protocoll instead of plain http, specify it here.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'HttpType'} = 'http';</code>

B.1.1.6. ScriptAlias

Description	Value
Description:	Prefix to index.pl, like it is specified in your webserver configuration.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'ScriptAlias'} = 'otrs/';</code>

B.1.1.7. AdminEmail

Description	Value
Description:	Emailaddress of the system administrator.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'AdminEmail'} = 'admin@example.com';</code>

B.1.1.8. Organization

Description	Value
Description:	If an organization or company name is specified, every outgoing message will contain a X-header entry with this string.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Organization'} = 'Example Company';</code>

B.1.1.9. DefaultCharset

Description	Value
Description:	Default frontend charset. "utf-8" is a good choice for environments with many possible charsets. Specify another charset (e.g "iso-8859-1"), if only this specific charset is needed.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'DefaultCharset'} = 'iso-8859-1';</code>

B.1.1.10. DefaultLanguage

Description	Value
Description:	The default frontend language. Possible values are bg, cz, de, en, el, es, fi, fr, hu, it, nb, nl, pl, pt, ru, th and sv.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'DefaultLanguage'} = 'en';</code>

B.1.1.11. DefaultUsedLanguages

Description	Value
Description:	Shortname for languages (short name = long name and file).
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'DefaultUsedLanguages'} = { 'pt' => 'Portugu&ecirc;s', 'sk_SK' => 'Slovak', 'es' => 'Espa&ntilde;ol', 'da' => 'Dansk', 'ru' => 'Russian', 'fr' => 'Fran&ccedil;ais', 'bg' => 'Bulgarian', 'sv' => 'Svenska', 'it' => 'Italiano', 'hu' => 'Hungarian', 'zh_CN' => 'Simplified Chinese', 'nb_NO' => 'Norsk bokm&aring;l', 'pl' => 'Polski', 'cz' => 'Czech', 'nl' => 'Nederlands', 'fi' => 'Suomi', 'en' => 'English', 'de' => 'Deutsch', 'el' => 'Greek', 'pt_BR' => 'Portugu&ecirc;s Brasileiro' }; </pre>

B.1.1.12. DefaultTheme

Description	Value
Description:	Default frontend HTML theme (Standard or Lite).
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'DefaultTheme'} = 'Standard'; </pre>

B.1.1.13. DefaultTheme::HostBased

Description	Value
Description:	Individual themes for Customer and Agents are possible in OTRS. These can be activated by a seperate domain. The field "Key" contains the domain or a regexp to match a domin. In field "Content" the Theme will be insert.
Group:	Framework
SubGroup:	Core
Valid:	0

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'DefaultTheme::HostBased'} = { 'host2\\example\\.com' => 'SomeTheme2', 'host1\\example\\.com' => 'SomeTheme1' };</pre>

B.1.1.14. CheckMXRecord

Description	Value
Description:	Check the MX record of email addresses before sending a mail?
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CheckMXRecord'} = '1';</pre>

B.1.1.15. CheckEmailAddresses

Description	Value
Description:	Check the syntax of email addresses?
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CheckEmailAddresses'} = '1';</pre>

B.1.1.16. CheckEmailValidAddress

Description	Value
Description:	Regex for email addresses, that aren't syntactically valid but necessary for the system, e.g. "root@localhost".
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CheckEmailValidAddress'} = '^(root@localhost admin@localhost)';</pre>

B.1.1.17. CheckEmailInvalidAddress

Description	Value
Description:	Regex of not allowed email addresses.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CheckEmailInvalidAddress'} = '@(aa aaa aaaa aaaaa abc any any</code>

B.1.1.18. CGILogPrefix

Description	Value
Description:	Log prefix for the CGI scripts of the system.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CGILogPrefix'} = 'OTRS-CGI';</code>

B.1.1.19. DemoSystem

Description	Value
Description:	Activate the demo system? If set to "Yes", agents can change their preferences like selection of language and theme via the agent frontend just for the current session. Also it isn't possible to change passwords.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'DemoSystem'} = '0';</code>

B.1.1.20. SwitchToUser

Description	Value
Description:	Allow the admin to switch into a selected user session.
Group:	Framework

Description	Value
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SwitchToUser'} = '0';</code>

B.1.1.21. NotificationSenderName

Description	Value
Description:	Sender name for system notification messages.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'NotificationSenderName'} = 'OTRS Notification Master';</code>

B.1.1.22. NotificationSenderEmail

Description	Value
Description:	Email address of sender for notification messages.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'NotificationSenderEmail'} = 'otrs@<OTRS_CONFIG_FQDN>';</code>

B.1.1.23. System::Permission

Description	Value
Description:	The different system permissions of the agents in the system (also possible in addition is note, close, pending, customer, freetext, forward and bounce - rw should be the latest in the line).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'System::Permission'} = ['ro', 'move_into', 'create', 'owner', 'priority', 'rw'];</code>

B.1.1.24. System::Customer::Permission

Description	Value
Description:	The different system permissions for customers.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'System::Customer::Permission'} = ['ro', 'rw'];</code>

B.1.2. Core::Log

B.1.2.1. LogModule

Description	Value
Description:	Log module for the system. "File" writes all messages in a given logfile, "SysLog" uses the syslog daemon of the system, e.g. syslogd.
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LogModule'} = 'Kernel::System::Log::SysLog';</code>

B.1.2.2. LogModule::SysLog::Facility

Description	Value
Description:	If "SysLog" was selected for LogModule, you can specify a special log facility.
Group:	Framework
SubGroup:	Core::Log

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LogModule::SysLog::Facility'} = 'user';</code>

B.1.2.3. LogModule::SysLog::Charset

Description	Value
Description:	If "SysLog" was selected for LogModule, you can specify the charset that should be used for login.
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LogModule::SysLog::Charset'} = 'iso-8859-1';</code>

B.1.2.4. LogModule::LogFile

Description	Value
Description:	If "file" was selected for LogModule, you have to specify a logfile. The file will be created by the system, if it doesn't exist.
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LogModule::LogFile'} = '/tmp/otrs.log';</code>

B.1.2.5. LogModule::LogFile::Date

Description	Value
Description:	Would you like to add a time stamp as suffix for every log entry?
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LogModule::LogFile::Date'} = '0';</code>

B.1.2.6. LogSystemCacheSize

Description	Value
Description:	Cache size for admin system log (in KBytes).
Group:	Framework
SubGroup:	Core::Log
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'LogSystemCacheSize'} = '4096';</code>

B.1.3. Core::MIME-Viewer**B.1.3.1. MIME-Viewer###application/excel**

Description	Value
Description:	Path to converter to view Microsoft Excel files in the web interface.
Group:	Framework
SubGroup:	Core::MIME-Viewer
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'MIME-Viewer'}->{'application/excel'} = 'xlhtml';</code>

B.1.3.2. MIME-Viewer###application/msword

Description	Value
Description:	Path to converter to view Microsoft Word files in the web interface.
Group:	Framework
SubGroup:	Core::MIME-Viewer
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'MIME-Viewer'}->{'application/msword'} = 'wvWare';</code>

B.1.3.3. MIME-Viewer###application/pdf

Description	Value
Description:	Path to programm to view PDF Documents in the web interface.
Group:	Framework
SubGroup:	Core::MIME-Viewer
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'MIME-Viewer'}->{'application/pdf'} = 'pdftohtml -stdout -i';</code>

B.1.3.4. MIME-Viewer###text/xml

Description	Value
Description:	Path to converter to view XML files in the web interface.
Group:	Framework
SubGroup:	Core::MIME-Viewer
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'MIME-Viewer'}->{'text/xml'} = '<OTRS_CONFIG_Home>/scripts/too</code>

B.1.4. Core::Package

B.1.4.1. Package::RepositoryRoot

Description	Value
Description:	Location to get online repository list for additional packages. The first available result will be used.
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Package::RepositoryRoot'} = ['http://ftp.otrs.org/pub/otrs/misc/packages/repository.xml', 'http://otrs.org/repository.xml'];</code>

B.1.4.2. Package::RepositoryList

Description	Value
Description:	List of online repositorys.
Group:	Framework
SubGroup:	Core::Package
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Package::RepositoryList'} = { 'ftp://ftp.example.com/pub/otrs/misc/packages/' => '[Example] ftp://ftp };</pre>

B.1.4.3. Package::Timeout

Description	Value
Description:	http/ftp timeout for package downloads (in sec).
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Package::Timeout'} = '12';</pre>

B.1.4.4. Package::Proxy

Description	Value
Description:	Fetch packages via proxy.
Group:	Framework
SubGroup:	Core::Package
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Package::Proxy'} = 'http://proxy.sn.no:8001/';</pre>

B.1.5. Core::Sendmail**B.1.5.1. SendmailModule**

Description	Value
Description:	Module to send E-Mails. "Sendmail" directly uses the sendmail binary of your operating system. "SMTP" uses a specified mailserver.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SendmailModule'} = 'Kernel::System::Email::Sendmail';</code>

B.1.5.2. SendmailModule::CMD

Description	Value
Description:	If "Sendmail" was selected for SendmailModule, you have to specify the location of your sendmail binary and the needed options.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SendmailModule::CMD'} = '/usr/sbin/sendmail -i -f';</code>

B.1.5.3. SendmailModule::Host

Description	Value
Description:	If "SMTP" was selected for SendmailModule, you have to specify the mailhost, that sends out the mails.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SendmailModule::Host'} = 'mail.example.com';</code>

B.1.5.4. SendmailModule::Port

Description	Value
-------------	-------

Description	Value
Description:	If "SMTP" was selected for SendmailModule, specify the port where your mailserver is listening for incoming connections.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'SendmailModule::Port'} = '25';</code>

B.1.5.5. SendmailModule::AuthUser

Description	Value
Description:	If "SMTP" was selected for SendmailModule and you need to authenticate to your mailserver, specify the username.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'SendmailModule::AuthUser'} = 'MailserverLogin';</code>

B.1.5.6. SendmailModule::AuthPassword

Description	Value
Description:	If "SMTP" was selected for SendmailModule and you need to authenticate to your mailserver, specify the password.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'SendmailModule::AuthPassword'} = 'MailserverPassword';</code>

B.1.5.7. SendmailBcc

Description	Value
-------------	-------

Description	Value
Description:	Send all outgoing email via bcc to the specified address. Please use this only for backup reasons!
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'SendmailBcc'} = '';</code>

B.1.5.8. SendmailNotificationEnvelopeFrom

Description	Value
Description:	If set, this address is used as envelope from header in outgoing notifications. If no address is specified, the envelope from header is empty.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'SendmailNotificationEnvelopeFrom'} = '';</code>

B.1.6. Core::Session

B.1.6.1. SessionModule

Description	Value
Description:	Module to store the session data. Advantage of "DB" is that you can split the frontend server from the db server. "FS" or "IPC" is much faster.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionModule'} = 'Kernel::System::AuthSession::DB';</code>

B.1.6.2. SessionName

Description	Value
-------------	-------

Description	Value
Description:	Name of the session key. E.g. Session, SessionID or OTRS.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionName'} = 'Session';</code>

B.1.6.3. SessionCheckRemoteIP

Description	Value
Description:	If the application is used for example via a proxy farm or a dialup connection, the remote ip address is mostly different for the requests. Turn of this check, if you are in this situation.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionCheckRemoteIP'} = '1';</code>

B.1.6.4. SessionDeleteIfNotRemoteID

Description	Value
Description:	Delete a session, if the session id is used with an invalid remote IP?
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionDeleteIfNotRemoteID'} = '1';</code>

B.1.6.5. SessionMaxTime

Description	Value
Description:	Maximal valid time for a session id (in seconds).
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionMaxTime'} = '36000';</code>

B.1.6.6. SessionMaxIdleTime

Description	Value
Description:	A session will be killed and the user will be logged out after this time of inactivity (in seconds).
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionMaxIdleTime'} = '18000';</code>

B.1.6.7. SessionDeleteIfTimeToOld

Description	Value
Description:	Delete requested sessions if they are timed out?
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionDeleteIfTimeToOld'} = '1';</code>

B.1.6.8. SessionUseCookie

Description	Value
Description:	Should the session management use html cookies? If html cookies are disabled or if the client browser disabled html cookies, then the system will work as usual and append the session id to the links.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionUseCookie'} = '1';</code>

B.1.6.9. SessionUseCookieAfterBrowserClose

Description	Value
Description:	Store cookies after the browser has ben closed?
Group:	Framework
SubGroup:	Core::Session

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionUseCookieAfterBrowserClose'} = '0';</code>

B.1.6.10. SessionDir

Description	Value
Description:	If "FS" was selected for SessionModule you have to specify a directory where the session data will be stored.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionDir'} = '<OTRS_CONFIG_Home>/var/sessions';</code>

B.1.6.11. SessionTable

Description	Value
Description:	If "DB" was selected for SessionModule, you have to specify a table in database where session data will be stored.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionTable'} = 'sessions';</code>

B.1.6.12. SessionTableID

Description	Value
Description:	If "DB" was selected for SessionModule, you have to specify a column for ID in session table.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionTableID'} = 'session_id';</code>

B.1.6.13. SessionTableValue

Description	Value
Description:	If "DB" was selected for SessionModule, you have to specify a column for the values in session table.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionTableValue'} = 'session_value';</code>

B.1.6.14. CustomerPanelSessionName

Description	Value
Description:	Name of the key for customer sessions.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelSessionName'} = 'CSID';</code>

B.1.7. Core::SpellChecker**B.1.7.1. SpellChecker**

Description	Value
Description:	Enable or disable spell checker support.
Group:	Framework
SubGroup:	Core::SpellChecker
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SpellChecker'} = '1';</code>

B.1.7.2. SpellCheckerBin

Description	Value
-------------	-------

Description	Value
Description:	Install ispell or aspell on your system, if you want to use a spellchecker. Please specify the path to the aspell or ispell binary on your operating system.
Group:	Framework
SubGroup:	Core::SpellChecker
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SpellCheckerBin'} = '/usr/bin/ispell';</code>

B.1.7.3. SpellCheckerDictDefault

Description	Value
Description:	Default spell checker dictionary.
Group:	Framework
SubGroup:	Core::SpellChecker
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SpellCheckerDictDefault'} = 'english';</code>

B.1.7.4. SpellCheckerIgnore

Description	Value
Description:	A list of words, that are ignored by the spell checker by default.
Group:	Framework
SubGroup:	Core::SpellChecker
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SpellCheckerIgnore'} = ['www', 'webmail', 'https', 'http', 'html', 'rfc'];</code>

B.1.8. Core::Time

B.1.8.1. TimeInputFormat

Description	Value
-------------	-------

Description	Value
Description:	Used date input format in forms (option or input fields).
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeInputFormat'} = 'Option';</code>

B.1.8.2. TimeCalendarLookup

Description	Value
Description:	Enable a calender lookup window on date selections.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeCalendarLookup'} = '1';</code>

B.1.8.3. TimeShowAlwaysLong

Description	Value
Description:	Show time in long (days, hours, minutes) or short (days, hours) format.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeShowAlwaysLong'} = '0';</code>

B.1.8.4. TimeZone

Description	Value
Description:	Set the system time zone (required a system with GMT as system time). Otherwise this is a diff time to the local time.
Group:	Framework
SubGroup:	Core::Time
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TimeZone'} = '+0';</code>

B.1.8.5. TimeVacationDays

Description	Value
Description:	Add your permanent vacation days.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'TimeVacationDays'} = { '1' => { '1' => 'New Year\'s Eve!' }, '12' => { '25' => 'First Christmas Day', '24' => 'Christmas', '26' => 'Second Christmas Day', '31' => 'Silvester' }, '5' => { '1' => '1 St. May' } }; </pre>

B.1.8.6. TimeVacationDaysOneTime

Description	Value
Description:	One time vacation days.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'TimeVacationDaysOneTime'} = { '2004' => { '1' => { '1' => 'test' } } }; </pre>

B.1.8.7. TimeWorkingHours

Description	Value
Description:	Hours and week days to count the working time.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre> \$Self->{'TimeWorkingHours'} = { 'Mon' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Tue' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Fri' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Wed' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Sun' => [], 'Sat' => [], 'Thu' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'] }; </pre>

B.1.9. Core::Web

B.1.9.1. DefaultViewNewLine

Description	Value
Description:	Automated line break in text messages after that count of characters.

Description	Value
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'DefaultViewNewLine'} = '85';</code>

B.1.9.2. DefaultPreViewLines

Description	Value
Description:	Count of lines that are displayed in the preview for messages (e.g. for tickets in the QueueView).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'DefaultPreViewLines'} = '18';</code>

B.1.9.3. DefaultViewLines

Description	Value
Description:	Count of lines that are displayed in text messages (e.g. ticket lines in the QueueZoom).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'DefaultViewLines'} = '6000';</code>

B.1.9.4. AttachmentDownloadType

Description	Value
Description:	Show the attachments of a ticket in the browser (inline) or just make them downloadable (attachment).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'AttachmentDownloadType'} = 'attachment';</code>

B.1.9.5. WebMaxFileUpload

Description	Value
Description:	Maximal size for file uploads via the browser (in Bytes).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'WebMaxFileUpload'} = '5242880';</code>

B.1.9.6. WebUploadCacheModule

Description	Value
Description:	Select the module to handle uploads via the web interface. "DB" stores all uploads in the database, "FS" uses the file system.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'WebUploadCacheModule'} = 'Kernel::System::Web::UploadCache::D</code>

B.1.9.7. Frontend::Output::PostFilter###ActiveElementFilter

Description	Value
Description:	A filter for html output of the application. With this filter you can for example filter out javascript or suppress java applets.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Frontend::Output::PostFilter'}->{'ActiveElementFilter'} = { 'Debug' => '0', 'Module' => 'Kernel::Output::HTML::OutputFilterActiveElement' };</code>

B.1.10. Crypt::PGP

B.1.10.1. PGP

Description	Value
Description:	Enable PGP support?
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PGP'} = '0';</code>

B.1.10.2. PGP::Bin

Description	Value
Description:	Path to PGP binary.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PGP::Bin'} = '/usr/bin/gpg';</code>

B.1.10.3. PGP::Options

Description	Value
Description:	Options for PGP binary.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PGP::Options'} = '--homedir /var/lib/wwrun/.gnupg/ --batch -</code>

B.1.10.4. PGP::Key::Password

Description	Value
Description:	Password for PGP key.

Description	Value
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'PGP::Key::Password'} = { '488A0B8F' => 'SomePassword', 'D2DF79FA' => 'SomePassword' };</pre>

B.1.11. Crypt::SMIME

B.1.11.1. SMIME

Description	Value
Description:	Enable S/MIME support?
Group:	Framework
SubGroup:	Crypt::SMIME
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'SMIME'} = '0';</pre>

B.1.11.2. SMIME::Bin

Description	Value
Description:	Path to openssl binary. Maybe openssl need a HOME env (\$ENV{HOME} = '/var/lib/wwwrun');!
Group:	Framework
SubGroup:	Crypt::SMIME
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'SMIME::Bin'} = '/usr/bin/openssl';</pre>

B.1.11.3. SMIME::CertPath

Description	Value
Description:	Directory where SSL certificates are stored.
Group:	Framework
SubGroup:	Crypt::SMIME

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SMIME::CertPath'} = '/etc/ssl/certs';</code>

B.1.11.4. SMIME::PrivatePath

Description	Value
Description:	Directory where private SSL certificates are stored.
Group:	Framework
SubGroup:	Crypt::SMIME
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SMIME::PrivatePath'} = '/etc/ssl/private';</code>

B.1.12. Frontend::Admin::ModuleRegistration

B.1.12.1. Frontend::Module###Admin

Description	Value
Description:	Frontend module registration for the Admin object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'Admin'} = { 'NavBar' => ['Prio' => '10000', 'Block' => 'ItemArea', 'Image' => 'admin.png', 'NavBar' => 'Admin', 'Type' => 'Menu', 'AccessKey' => 'a', 'Description' => 'Admin-Area', 'Name' => 'Admin', 'Link' => 'Action=Admin'], 'NavBarName' => 'Admin', 'Description' => 'Admin-Area', 'Title' => 'Admin', 'NavBarModule' => { 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin' }, 'Group' => [] }; </pre>

B.1.12.2. Frontend::Module###AdminUser

Description	Value
Description:	Frontend module registration for the AdminUser object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminUser'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'User', 'NavBarModule' => { 'Prio' => '100', 'Block' => 'Block1', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Users' }, 'Group' => [] }; </pre>

B.1.12.3. Frontend::Module###AdminGroup

Description	Value
Description:	Frontend module registration for the AdminGroup object in the admin area.
Group:	Framework

Description	Value
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminGroup'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Group', 'NavBarModule' => { 'Prio' => '150', 'Block' => 'Block1', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Groups' }, 'Group' => ['admin'] }; </pre>

B.1.12.4. Frontend::Module###AdminUserGroup

Description	Value
Description:	Frontend module registration for the AdminUserGroup object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminUserGroup'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Users <-> Groups', 'NavBarModule' => { 'Prio' => '200', 'Block' => 'Block1', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Users <-> Groups' }, 'Group' => ['admin'] }; </pre>

B.1.12.5. Frontend::Module###AdminCustomerUser

Description	Value
Description:	Frontend module registration for the AdminCustomerUser object in the admin area.
Group:	Framework

Description	Value
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminCustomerUser'} = { 'NavBar' => ['Prio' => '9000', 'Image' => 'folder_yellow.png', 'NavBar' => 'Ticket', 'Type' => "", 'AccessKey' => 'c', 'Description' => 'Edit Customer Users', 'Name' => 'Customer', 'Link' => 'Action=AdminCustomerUser&Nav=Agent'], 'GroupRo' => ["], 'NavBarName' => "", 'Description' => 'Edit Customer Users', 'Title' => 'Customer User', 'NavBarModule' => { 'Prio' => '300', 'Block' => 'Block1', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Customer Users' }, 'Group' => ['admin', 'users']; </pre>

B.1.12.6. Frontend::Module###AdminCustomerUserGroup

Description	Value
Description:	Frontend module registration for the AdminCustomerUserGroup object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminCustomerUserGroup'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Customer Users <-> Groups', 'NavBarModule' => { 'Prio' => '400', 'Block' => 'Block1', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Customer Users <-> Groups' }, 'Group' => ['admin']; </pre>

B.1.12.7. Frontend::Module###AdminRole

Description	Value
Description:	Frontend module registration for the AdminRole object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminRole'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Role', 'NavBarModule' => { 'Prio' => '500', 'Block' => 'Block1', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Roles' }, 'Group' => ['admin']; </pre>

B.1.12.8. Frontend::Module###AdminRoleUser

Description	Value
Description:	Frontend module registration for the AdminRoleUser object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminRoleUser'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Roles <-> Users', 'NavBarModule' => { 'Prio' => '600', 'Block' => 'Block1', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Roles <-> Users' }, 'Group' => []; </pre>

B.1.12.9. Frontend::Module###AdminRoleGroup

Description	Value
Description:	Frontend module registration for the AdminRoleGroup object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminRoleGroup'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Roles <-> Groups', 'NavBarModule' => { 'Prio' => '700', 'Block' => 'Block1', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Roles <-> Groups' }, 'Group' => []; </pre>

B.1.12.10. Frontend::Module###AdminSMIME

Description	Value
Description:	Frontend module registration for the AdminSMIME object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1

Description	Value
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminSMIME'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'SMIME Management', 'NavBarModule' => { 'Prio' => '500', 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'SMIME' }, 'Group' => ['admin'] }; </pre>

B.1.12.11. Frontend::Module###AdminPGP

Description	Value
Description:	Frontend module registration for the AdminPGP object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminPGP'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'PGP Key Management', 'NavBarModule' => { 'Prio' => '600', 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'PGP' }, 'Group' => ['admin'] }; </pre>

B.1.12.12. Frontend::Module###AdminPOP3

Description	Value
Description:	Frontend module registration for the AdminPop3 object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminPOP3'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'POP3 Account', 'NavBarModule' => { 'Prio' => '100', 'Block' => 'Block4', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'PostMaster POP3 Account' }, 'Group' => [] }; </pre>

B.1.12.13. Frontend::Module###AdminPostMasterFilter

Description	Value
Description:	Frontend module registration for the AdminPostMasterFilter object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminPostMasterFilter'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'PostMaster Filter', 'NavBarModule' => { 'Prio' => '200', 'Block' => 'Block4', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'PostMaster Filter' }, 'Group' => [] }; </pre>

B.1.12.14. Frontend::Module###AdminEmail

Description	Value
Description:	Frontend module registration for the AdminEmail object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminEmail'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Admin-Email', 'NavBarModule' => { 'Prio' => '400', 'Block' => 'Block4', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Admin Notification' }, 'Group' => [] }; </pre>

B.1.12.15. Frontend::Module###AdminSession

Description	Value
Description:	Frontend module registration for the AdminSession object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminSession'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Session Management', 'NavBarModule' => { 'Prio' => '500', 'Block' => 'Block4', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Session Management' }, 'Group' => [] }; </pre>

B.1.12.16. Frontend::Module###AdminLog

Description	Value
Description:	Frontend module registration for the AdminLog object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminLog'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'System Log', 'NavBarModule' => { 'Prio' => '600', 'Block' => 'Block4', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'System Log' }, 'Group' => ['admin'] }; </pre>

B.1.12.17. Frontend::Module###AdminSelectBox

Description	Value
Description:	Frontend module registration for the AdminSelectBox object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminSelectBox'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Select box', 'NavBarModule' => { 'Prio' => '700', 'Block' => 'Block4', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Select Box' }, 'Group' => ['admin'] }; </pre>

B.1.12.18. Frontend::Module###AdminPackageManager

Description	Value
Description:	Frontend module registration for the AdminPackageManager object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminPackageManager'} = { 'NavBarName' => 'Admin', 'Description' => 'Software Package Manager', 'Title' => 'Package Manager', 'NavBarModule' => { 'Prio' => '1000', 'Block' => 'Block4', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Package Manager' }, 'Group' => [] };</pre>

B.1.13. Frontend::Agent

B.1.13.1. LoginURL

Description	Value
Description:	Alternate URL, where the login link refers to.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'LoginURL'} = 'http://host.example.com/login.html';</pre>

B.1.13.2. LogoutURL

Description	Value
Description:	Alternate URL, where the logout link refers to.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'LogoutURL'} = 'http://host.example.com/thanks-for-using-otrs.'</pre>

B.1.13.3. PreApplicationModule

Description	Value
Description:	This module is usefull to load some specific user options or to redirect not accept new application news. For example: Kernel::Modules::AgentWorkingRoleSelection or Kernel::Modules::AgentInfo
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'PreApplicationModule'} = 'Kernel::Modules::AgentInfo';</code>

B.1.13.4. InfoKey

Description	Value
Description:	Key to check with Kernel::Modules::AgentInfo module. If this user preferences key is true, then the message is accepted by the system.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'InfoKey'} = 'wpt22';</code>

B.1.13.5. InfoFile

Description	Value
Description:	File that is displayed, if located under Kernel/Output/HTML/Standard/AgentInfo.dtl.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'InfoFile'} = 'AgentInfo';</code>

B.1.13.6. LostPassword

Description	Value
Description:	Activate lost password feature fpr agents?
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LostPassword'} = '1';</code>

B.1.13.7. ShowMotd

Description	Value
Description:	Show message of the day after login screen?
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'ShowMotd'} = '0';</code>

B.1.13.8. NotificationSubjectLostPassword

Description	Value
Description:	Subject for the notification email to agents about new password.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'NotificationSubjectLostPassword'} = 'New OTRS Password!';</code>

B.1.13.9. NotificationBodyLostPassword

Description	Value
Description:	Body for notification mail to agents about new password.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self->{'NotificationBodyLostPassword'} = 'Hi <OTRS_USERFIRSTNAME>, you or someone impersonating you has requested to change your OTRS password. New Password: <OTRS_NEWPW> <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/<OTRS_CONFIG_ScriptAlias>in Your OTRS Notification Master ';</pre>

B.1.13.10. Frontend::NavBarStyle

Description	Value
Description:	With this setting you can define how the icons in the NavBar are displayed.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Frontend::NavBarStyle'} = 'Classic';</pre>

B.1.13.11. Frontend::NavBarStyle::ShowSelectedArea

Description	Value
Description:	Do you want to highlight the link to the selected module in the NavBar?
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Frontend::NavBarStyle::ShowSelectedArea'} = '0';</pre>

B.1.14. Frontend::Agent::Auth::LDAP

B.1.14.1. UserSyncLDAPMap

Description	Value
Description:	If "LDAP" was selected for AuthModule, you can specify attributes for user LDAP sync on login.
Group:	Framework

Description	Value
SubGroup:	Frontend::Agent::Auth::LDAP
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'UserSyncLDAPMap'} = { 'Email' => 'mail', 'Lastname' => 'sn', 'Firstname' => 'givenName' };</pre>

B.1.14.2. UserSyncLDAPGroups

Description	Value
Description:	If "LDAP" was selected for AuthModule, you can specify initial user groups for first login.
Group:	Framework
SubGroup:	Frontend::Agent::Auth::LDAP
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'UserSyncLDAPGroups'} = ['users'];</pre>

B.1.15. Frontend::Agent::ModuleNotify

B.1.15.1. Frontend::NotifyModule###1-CharsetCheck

Description	Value
Description:	Module to inform agents via the agent interface about the used charset. A notification is displayed, if the default charset is not used, e.g. in tickets.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Frontend::NotifyModule'}->{'1-CharsetCheck'} = { 'Module' => 'Kernel::Output::HTML::NotificationCharsetCheck' };</pre>

B.1.15.2. Frontend::NotifyModule###2-UID-Check

Description	Value
Description:	Module to display a notification in the agent interface, if the system is used by the admin user (normally you shouldn't work as admin).
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Frontend::NotifyModule'}->{'2-UID-Check'} = { 'Module' => 'Kernel::Output::HTML::NotificationUIDCheck' };</pre>

B.1.15.3. Frontend::NotifyModule###3-ShowAgentOnline

Description	Value
Description:	Module to show currently logged in agents in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::NotifyModule'}->{'3-ShowAgentOnline'} = { 'Module' => 'Kernel::Output::HTML::NotificationAgentOnline' };</pre>

B.1.15.4. Frontend::NotifyModule###4-ShowCustomerOnline

Description	Value
Description:	Module to show currently logged in customers in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::NotifyModule'}->{'4-ShowCustomerOnline'} = { 'Module' => 'Kernel::Output::HTML::NotificationCustomerOnline' };</pre>

B.1.16. Frontend::Agent::ModuleRegistration

B.1.16.1. Frontend::Module###Logout

Description	Value
Description:	Frontend module registration for the Logout object in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'Logout'} = { 'NavBar' => [{ 'Prio' => '100', 'Block' => 'ItemPre', 'Image' => 'exit.png', 'NavBar' => "", 'AccessKey' => 'l', 'Description' => 'Logout', 'Name' => 'Logout', 'Link' => 'Action=Logout' }], 'NavBarName' => "", 'Description' => 'Logout', 'Title' => "" }; </pre>

B.1.16.2. Frontend::Module###AgentPreferences

Description	Value
Description:	Frontend module registration for the AgentPreference object in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentPreferences'} = { 'NavBar' => [{ 'Prio' => '1000', 'Block' => "", 'Image' => 'prefer.png', 'NavBar' => "", 'Type' => "", 'AccessKey' => 'p', 'Description' => 'Agent Preferences', 'Name' => 'Preferences', 'Link' => 'Action=AgentPreferences' }], 'NavBarName' => "", 'Description' => 'Agent Preferences', 'Title' => 'Preferences' }; </pre>

B.1.16.3. Frontend::Module###AgentSpelling

Description	Value
Description:	Frontend module registration for the AgentSpelling object in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentSpelling'} = { 'NavBarName' => "", 'Description' => 'Spell checker', 'Title' => 'Spell Checker' }; </pre>

B.1.16.4. Frontend::Module###AgentBook

Description	Value
Description:	Frontend module registration for the AgentBook object in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentBook'} = { 'NavBarName' => "", 'Description' => 'Address book of CustomerUser sources', 'Title' => 'Address Book' }; </pre>

B.1.16.5. Frontend::Module###AgentLookup

Description	Value
Description:	Frontend module registration for the AgentLookup object in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentLookup'} = { 'NavBarName' => "", 'Description' => 'Data table lookup module.', 'Title' => 'Lookup' };</pre>

B.1.16.6. Frontend::Module###AgentLinkObject

Description	Value
Description:	Frontend module registration for the AgentLinkObject object in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentLinkObject'} = { 'NavBarName' => "", 'Description' => 'Link Object', 'Title' => 'Link Object' };</pre>

B.1.16.7. Frontend::Module###AgentInfo

Description	Value
Description:	Frontend module registration for the AgentInfo object in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentInfo'} = { 'NavBarName' => "", 'Description' => 'Generic Info module', 'Title' => 'Info' };</pre>

B.1.16.8. Frontend::Module###AgentCalendarSmall

Description	Value
Description:	Frontend module registration for the AgentCalendarSmall object in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentCalendarSmall'} = { 'NavBarName' => "", 'Description' => 'Small calendar for date selection.', 'Title' => 'Calendar' };</pre>

B.1.16.9. Frontend::Module###SystemStats

Description	Value
Description:	Frontend module registration for the SystemStats object in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'SystemStats'} = { 'NavBar' => [{ 'Prio' => '400', 'Block' => "", 'Image' => 'stats.png', 'NavBar' => 'Ticket', 'Type' => "", 'AccessKey' => 't', 'Description' => 'Stats-Area', 'Name' => 'Stats', 'Link' => 'Action=SystemStats' }], 'GroupRo' => ['stats'], 'NavBarName' => 'Ticket', 'Description' => 'Stats', 'Title' => 'Stats' };</pre>

B.1.17. Frontend::Agent::Preferences

B.1.17.1. PreferencesTableValue

Description	Value
Description:	Name of column in preferences table to store the data.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PreferencesTableValue'} = 'preferences_value';</code>

B.1.17.2. PreferencesTableUserID

Description	Value
Description:	Name of column in preference table, where the UserID is stored.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PreferencesTableUserID'} = 'user_id';</code>

B.1.17.3. PreferencesView

Description	Value
Description:	Display order of the different items in the preference view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PreferencesView'} = ['Frontend', 'Mail Management', 'Other Options'];</code>

B.1.17.4. PreferencesGroups###Password

Description	Value
-------------	-------

Description	Value
Description:	Parameters for the Password object in the preference view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'Password'} = { 'Prio' => '1000', 'PasswordNeedDigit' => '0', 'Label' => 'Change Password', 'PasswordMin2Characters' => '0', 'PasswordRegExp' => "", 'Module' => 'Kernel::Output::HTML::PreferencesPassword', 'PasswordMin2Lower2UpperCharacters' => '0', 'Area' => 'Agent', 'Activ' => '1', 'Colum' => 'Other Options', 'PasswordMinSize' => '0' }; </pre>

B.1.17.5. PreferencesGroups###SpellDict

Description	Value
Description:	Parameters for the SpellDict object in the preference view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'SpellDict'} = { 'Prio' => '5000', 'Label' => 'Spelling Dictionary', 'PrefKey' => 'UserSpellDict', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' => 'Select your default spelling dictionary.', 'Activ' => '1', 'Data' => { 'deutsch' => 'Deutsch', 'english' => 'English' }, 'Colum' => 'Other Options' }; </pre>

B.1.17.6. PreferencesGroups###Comment

Description	Value
Description:	Parameters for the Comment object in the preference view.
Group:	Framework

Description	Value
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'Comment'} = { 'Prio' => '6000', 'Block' => 'Input', 'Label' => 'Comment', 'PrefKey' => 'UserComment', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' => 'Comment', 'Activ' => '0', 'Data' => '\$Env{"UserComment"}', 'Colum' => 'Other Options' }; </pre>

B.1.17.7. PreferencesGroups###FreeText

Description	Value
Description:	Parameters for the Freetext object in the preference view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'FreeText'} = { 'Prio' => '7000', 'Block' => 'Input', 'Label' => 'Comment', 'PrefKey' => 'UserFreeText', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' => 'Example for free text', 'Activ' => '1', 'Data' => '\$Env{"UserFreeText"}', 'Colum' => 'Other Options' }; </pre>

B.1.17.8. PreferencesGroups###Language

Description	Value
Description:	Parameters for the Language object in the preference view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'Language'} = { 'Prio' => '1000', 'Label' => 'Language', 'PrefKey' => 'UserLanguage', 'Module' => 'Kernel::Output::HTML::PreferencesLanguage', 'Desc' => 'Select your frontend language.', 'Activ' => '1', 'Colum' => 'Frontend' };</pre>

B.1.17.9. PreferencesGroups###Theme

Description	Value
Description:	Parameters for the Time object in the preference view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'Theme'} = { 'Prio' => '2000', 'Label' => 'Theme', 'PrefKey' => 'UserTheme', 'Module' => 'Kernel::Output::HTML::PreferencesTheme', 'Desc' => 'Select your frontend Theme.', 'Activ' => '1', 'Colum' => 'Frontend' };</pre>

B.1.18. Frontend::Customer

B.1.18.1. CustomerPanelUserID

Description	Value
Description:	UserID for customer panel.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerPanelUserID'} = '1';</pre>

B.1.18.2. CustomerGroupSupport

Description	Value
Description:	Activate support for customer groups?
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerGroupSupport'} = '0';</code>

B.1.18.3. CustomerGroupAlwaysGroups

Description	Value
Description:	If CustomerGroupSupport is enabled and you don't want to manage every user for this groups, then put the groups for all customer user in there.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerGroupAlwaysGroups'} = ['users', 'info'];</code>

B.1.18.4. CustomerPanelLoginURL

Description	Value
Description:	Alternate login URL for the customer panel..
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'CustomerPanelLoginURL'} = 'http://host.example.com/cgi-bin/lo</code>

B.1.18.5. CustomerPanelLogoutURL

Description	Value
Description:	Alternate logout URL for the customer panel.
Group:	Framework
SubGroup:	Frontend::Customer

Description	Value
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'CustomerPanelLogoutURL'} = 'http://host.example.com/cgi-bin/1</code>

B.1.18.6. CustomerPanelPreApplicationModule

Description	Value
Description:	CustomerPanelPreApplicationModule. This module and its PreRun() function will be executed, if defined, for every request. This module is useful to check some user options or to display news about new applications.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'CustomerPanelPreApplicationModule'} = 'Kernel::Modules::Custo</code>

B.1.18.7. CustomerPanel::InfoKey

Description	Value
Description:	Key to check with CustomerAccept. If this user preferences key is true, then the message is accepted by the system.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'CustomerPanel::InfoKey'} = 'CustomerAccept1';</code>

B.1.18.8. CustomerPanel::InfoFile

Description	Value
Description:	Shown InfoFile, that is located under Kernel/Output/HTML/Standard/CustomerAccept.dtl.
Group:	Framework

Description	Value
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'CustomerPanel::InfoFile'} = 'CustomerAccept';</code>

B.1.18.9. CustomerPanelLostPassword

Description	Value
Description:	Activate lost password feature for customers?
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelLostPassword'} = '1';</code>

B.1.18.10. CustomerPanelCreateAccount

Description	Value
Description:	Can Customers create their own account?
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelCreateAccount'} = '1';</code>

B.1.18.11. CustomerPanelSubjectLostPassword

Description	Value
Description:	Subject of notification mail to customers about new password.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelSubjectLostPassword'} = 'New OTRS Password!';</code>

B.1.18.12. CustomerPanelBodyLostPassword

Description	Value
Description:	Body of notification mail for customers about new password.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerPanelBodyLostPassword'} = 'Hi <OTRS_USERFIRSTNAME>, you or someone impersonating you has requested to change your OTRS password. New Password: <OTRS_NEWPW> <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/<OTRS_CONFIG_ScriptAlias>cu Your OTRS Notification Master ';</pre>

B.1.18.13. CustomerPanelSubjectNewAccount

Description	Value
Description:	Subject for notification email to customers about new account.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerPanelSubjectNewAccount'} = 'New OTRS Account!';</pre>

B.1.18.14. CustomerPanelBodyNewAccount

Description	Value
Description:	Body of notification email to customer about new account.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self->{'CustomerPanelBodyNewAccount'} = 'Hi <OTRS_USERFIRSTNAME>, you or someone impersonating you has created a new OTRS account for you (<OTRS_USERFIRSTNAME> <OTRS_USERLASTNAME>). Login: <OTRS_USERLOGIN> Password: <OTRS_USERPASSWORD> <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/<OTRS_CONFIG_ScriptAlias>cu Your OTRS Notification Master ';</pre>

B.1.19. Frontend::Customer::Auth

B.1.19.1. Customer::AuthModule

Description	Value
Description:	Module to authenticate customers.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::DB';</pre>

B.1.19.2. Customer::AuthModule::DB::Table

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, you have to specify the name of the table where your customer data are stored.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::DB::Table'} = 'customer_user';</pre>

B.1.19.3. Customer::AuthModule::DB::CustomerKey

Description	Value
-------------	-------

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, you have to specify the name of the column for the CustomerKey in your customer table.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::CustomerKey'} = 'login';</code>

B.1.19.4. Customer::AuthModule::DB::CustomerPassword

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, you have to specify the column name for the CustomerPassword in your customer table.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::CustomerPassword'} = 'pw';</code>

B.1.19.5. Customer::AuthModule::DB::DSN

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, you hav to specify the DSN for the connection to the customer table.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::DSN'} = 'DBI:mysql:database=custome</code>

B.1.19.6. Customer::AuthModule::DB::User

Description	Value
-------------	-------

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, you can specify a username to connect to your customer table.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::User'} = 'some_user';</code>

B.1.19.7. Customer::AuthModule::DB::Password

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, you can specify a password to connect to your customer table.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::Password'} = 'some_password';</code>

B.1.19.8. Customer::AuthModule::LDAP::Host

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, you can specify your LDAP host.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::Host'} = 'ldap.example.com';</code>

B.1.19.9. Customer::AuthModule::LDAP::BaseDN

Description	Value
-------------	-------

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, specify your BaseDN.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';</code>

B.1.19.10. Customer::AuthModule::LDAP::UID

Description	Value
Description:	If "LDAP" was selected for Customer::Authmodule, specefy the UID.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::UID'} = 'uid';</code>

B.1.19.11. Customer::AuthModule::LDAP::GroupDN

Description	Value
Description:	I "LDAP" was selected for Customer::Authmodule, you can check if the user is allowed to auth because he is in a posixGroup, e.g. user needs to be in a group xyz to use otrs. Specify the group, who may access the system.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::GroupDN'} = 'cn=otrsallow,ou=posix';</code>

B.1.19.12. Customer::AuthModule::LDAP::AccessAttr

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, you can specify access attributes here.

Description	Value
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::AccessAttr'} = 'memberUid';</code>

B.1.19.13. Customer::AuthModule::LDAP::UserAttr

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, you can specify user attributes. For LDAP posixGroups objectclass UID, for non ldap posixGroups objectclass full user DN.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::UserAttr'} = 'UID';</code>

B.1.19.14. Customer::AuthModule::LDAP::SearchUserDN

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule and your users have only anonymous access to the LDAP tree, but you want to search through the data, you can do this with a user who has access to the LDAP directory. Specify the username for this special user here.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::SearchUserDN'} = 'cn=binduser,ou=</code>

B.1.19.15. Customer::AuthModule::LDAP::SearchUserPw

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule and your users have only anonymous access to the LDAP tree, but you want to search through the data, you can do this with a user who has access to the LDAP directory. Specify the password for this special user here.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::SearchUserPw'} = 'some_password';</code>

B.1.19.16. Customer::AuthModule::LDAP::AlwaysFilter

Description	Value
Description:	If "LDAP" was selected, you can add a filter to each LDAP query, e.g. (mail=*), (objectclass=user) or (!objectclass=computer).
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::AlwaysFilter'} = '(!objectclass=computer)';</code>

B.1.19.17. Customer::AuthModule::LDAP::UserSuffix

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule and if you want to add a suffix to every customer login name, specify it here, e. g. you just want to write the username user but in your ldap directory exists user@domain.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::UserSuffix'} = '@domain.com';</code>

B.1.19.18. Customer::AuthModule::LDAP::Params

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule and special paramaters are needed for the Net::LDAP perl module, you can specify them here. See "perldoc Net::LDAP" for more information about the parameters.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::LDAP::Params'} = { 'async' => '0', 'timeout' => '120', 'version' => '3', 'port' => '389' };</pre>

B.1.19.19. Customer::AuthModule::Radius::Host

Description	Value
Description:	If "Radius" was selected for Customer::AuthModule, specify your radius host.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::Radius::Host'} = 'radiushost';</pre>

B.1.19.20. Customer::AuthModule::Radius::Password

Description	Value
Description:	If "Radius" was selected for Customer::AuthModule, specfiy the password to authenticate to your radius host.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::Radius::Password'} = 'radiussecret';</pre>

B.1.20. Frontend::Customer::ModuleNotify

B.1.20.1. CustomerFrontend::NotifyModule###1-ShowAgentOnline

Description	Value
Description:	Module that shows the currently logged in agents in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::NotifyModule'}->{'1-ShowAgentOnline'} = { 'Module' => 'Kernel::Output::HTML::NotificationAgentOnline' };</pre>

B.1.20.2. CustomerFrontend::NotifyModule###1-ShowCustomerOnline

Description	Value
Description:	Module that shows the currently logged in customers in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::NotifyModule'}->{'1-ShowCustomerOnline'} = 'Module' => 'Kernel::Output::HTML::NotificationCustomerOnline' };</pre>

B.1.21. Frontend::Customer::ModuleRegistration

B.1.21.1. CustomerFrontend::Module###Logout

Description	Value
Description:	Frontend module registration for the Logout object in the customer frontend.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1

Description	Value
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerFrontend::Module'}->{'Logout'} = { 'NavBar' => ['Prio' => '10', 'Image' => 'exit.png', 'NavBar' => "", 'AccessKey' => 'l', 'Description' => 'Logout', 'Name' => 'Logout', 'Link' => 'Action=Logout'], 'NavBarName' => "", 'Description' => 'Logout of customer panel', 'Title' => "" }; </pre>

B.1.21.2. CustomerFrontend::Module###CustomerPreferences

Description	Value
Description:	Frontend module registration for the CustomerPreferences object in the customer frontend.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerFrontend::Module'}->{'CustomerPreferences'} = { 'NavBar' => ['Prio' => '1000', 'Image' => 'prefer.png', 'NavBar' => "", 'AccessKey' => 'p', 'Description' => 'Preferences', 'Name' => 'Preferences', 'Link' => 'Action=CustomerPreferences'], 'NavBarName' => "", 'Description' => 'Customer preferences', 'Title' => 'Preferences' }; </pre>

B.1.21.3. CustomerFrontend::Module###CustomerCalendarSmall

Description	Value
Description:	Frontend module registration for the CustomerCalendarSmall object in the agent interface.

Description	Value
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}->{'CustomerCalendarSmall'} = { 'NavBarName' => "", 'Description' => 'Small calendar for date selection.', 'Title' => 'Calendar' };</pre>

B.1.21.4. CustomerFrontend::Module###CustomerAccept

Description	Value
Description:	Frontend module registration for the CustomerAccept object in the customer frontend.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}->{'CustomerAccept'} = { 'NavBarName' => "", 'Description' => 'To accept login infos', 'Title' => 'Info' };</pre>

B.1.22. Frontend::Customer::Preferences

B.1.22.1. PreferencesTable

Description	Value
Description:	The name of the table, where the customer preferences are stored.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'PreferencesTable'} = 'user_preferences';</pre>

B.1.22.2. PreferencesTableKey

Description	Value
Description:	Key for preferences table.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PreferencesTableKey'} = 'preferences_key';</code>

B.1.22.3. CustomerPreferences

Description	Value
Description:	Parameters for the customer preference table.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'CustomerPreferences'} = { 'Params' => { 'TableUserID' => 'user_id', 'TableValue' => 'preferences_value', 'TableKey' => 'preferences_key', 'Table' => 'customer_preferences' }, 'Module' => 'Kernel::System::CustomerUser::Preferences::DB' }; </pre>

B.1.22.4. CustomerPreferencesView

Description	Value
Description:	Order of the different items in the customer preferences view.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'CustomerPreferencesView'} = ['Frontend', 'Other Options']; </pre>

B.1.22.5. CustomerPreferencesGroups###Password

Description	Value
Description:	All parameters for the password area in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerPreferencesGroups'}->{'Password'} = { 'Prio' => '1000', 'PasswordNeedDigit' => '0', 'Label' => 'Change Password', 'PasswordMin2Characters' => '0', 'PasswordRegExp' => "", 'Module' => 'Kernel::Output::HTML::PreferencesPassword', 'PasswordMin2Lower2UpperCharacters' => '0', 'Area' => 'Customer', 'Activ' => '1', 'Colum' => 'Other Options', 'PasswordMinSize' => '0' }; </pre>

B.1.22.6. CustomerPreferencesGroups###Language

Description	Value
Description:	All parameters for the language area in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerPreferencesGroups'}->{'Language'} = { 'Prio' => '2000', 'Label' => 'Language', 'PrefKey' => 'UserLanguage', 'Module' => 'Kernel::Output::HTML::PreferencesLanguage', 'Desc' => 'Select your frontend language.', 'Activ' => '1', 'Colum' => 'Frontend' }; </pre>

B.1.22.7. CustomerPreferencesGroups###Theme

Description	Value
Description:	All parameters for the theme area in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences

Description	Value
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerPreferencesGroups'}->{'Theme'} = { 'Prio' => '1000', 'Label' => 'Theme', 'PrefKey' => 'UserTheme', 'Module' => 'Kernel::Output::HTML::PreferencesTheme', 'Desc' => 'Select your frontend Theme.', 'Activ' => '0', 'Colum' => 'Frontend' };</pre>

B.1.22.8. CustomerPreferencesGroups###PGP

Description	Value
Description:	All parameters for the PGP area in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerPreferencesGroups'}->{'PGP'} = { 'Prio' => '10000', 'Label' => 'PGP Key', 'PrefKey' => 'UserPGPKey', 'Module' => 'Kernel::Output::HTML::PreferencesPGP', 'Desc' => 'PGP Key Upload', 'Activ' => '1', 'Colum' => 'Other Options' }; </pre>

B.1.22.9. CustomerPreferencesGroups###SMIME

Description	Value
Description:	All parameters for the S/MIME area in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'CustomerPreferencesGroups'}->{'SMIME'} = { 'Prio' => '11000', 'Label' => 'SMIME Certificate', 'PrefKey' => 'UserSMIMEKey', 'Module' => 'Kernel::Output::HTML::PreferencesSMIME', 'Desc' => 'SMIME Certificate Upload', 'Activ' => '1', 'Colum' => 'Other Options' };</pre>

B.2. Ticket

B.2.1. Core::LinkObject

B.2.1.1. LinkObject###Ticket

Description	Value
Description:	Objects that can be linked.
Group:	Ticket
SubGroup:	Core::LinkObject
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'LinkObject'}->{'Ticket'} = { 'Type' => 'Object', 'LinkObjects' => ['Ticket', 'FAQ'], 'Name' => 'Ticket Object' };</pre>

B.2.2. Core::PostMaster

B.2.2.1. PostmasterMaxEmails

Description	Value
Description:	Maximal post master daemon email to own email-address a day. Loop-Protection! Default: 40
Group:	Ticket
SubGroup:	Core::PostMaster

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterMaxEmails'} = '40';</code>

B.2.2.2. PostMasterPOP3MaxEmailSize

Description	Value
Description:	Maximal size in KBytes for mails that can be fetched via pop3 (KBytes).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostMasterPOP3MaxEmailSize'} = '6144';</code>

B.2.2.3. PostMasterPOP3ReconnectMessage

Description	Value
Description:	"bin/PostMasterPOP3.pl" will reconnect to pop3 host after the specified count of messages.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostMasterPOP3ReconnectMessage'} = '20';</code>

B.2.2.4. LoopProtectionModule

Description	Value
Description:	Default loop protection modul
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LoopProtectionModule'} = 'Kernel::System::PostMaster::LoopPro</code>

B.2.2.5. LoopProtectionLog

Description	Value
Description:	If "FS" was selected for LoopProtectionModule, a logfile has to be specified.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LoopProtectionLog'} = '<OTRS_CONFIG_Home>/var/log/LoopProtect</code>

B.2.2.6. PostmasterAutoHTML2Text

Description	Value
Description:	Would you like to convert HTML mails into text messages?
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterAutoHTML2Text'} = '1';</code>

B.2.2.7. PostmasterFollowUpSearchInReferences

Description	Value
Description:	Would you like to execute followup checks on In-Reply-To or References headers for mails, that don't have a ticket number in the subject?
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterFollowUpSearchInReferences'} = '0';</code>

B.2.2.8. PostmasterUserID

Description	Value
Description:	Specify the postmaster DB userID.
Group:	Ticket

Description	Value
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterUserID'} = '1';</code>

B.2.2.9. PostmasterDefaultQueue

Description	Value
Description:	The postmaster default queue.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterDefaultQueue'} = 'Raw';</code>

B.2.2.10. PostmasterDefaultPriority

Description	Value
Description:	The default priority of new tickets.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterDefaultPriority'} = '3 normal';</code>

B.2.2.11. PostmasterDefaultState

Description	Value
Description:	Default state of new tickets.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterDefaultState'} = 'new';</code>

B.2.2.12. PostmasterFollowUpState

Description	Value
Description:	The state if a ticket got a followup.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterFollowUpState'} = 'open';</code>

B.2.2.13. PostmasterFollowUpOnUnlockAgentNotifyOnlyToOwner

Description	Value
Description:	Send agent follow up notification just to the owner, if a ticket is unlocked? The default is to send a notification to all agents, if a ticket is unlocked.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterFollowUpOnUnlockAgentNotifyOnlyToOwner'} = '0';</code>

B.2.2.14. PostmasterX-Header

Description	Value
Description:	All X-headers that should be scanned.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre> \$Self->{'PostmasterX-Header'} = ['From', 'To', 'Cc', 'Reply-To', 'ReplyTo', 'Subject', 'Message-ID', 'Message-Id', 'Resent-To', 'Resent-From', 'Precedence', 'Mailing-List', 'List-Id', 'List-Archive', 'Errors-To', 'References', 'In-Reply-To', 'X-Loop', 'X-Spam-Flag', 'X-Spam-Status', 'X-Spam-Level', 'X-No-Loop', 'X-Priority', 'Importance', 'X-Mailer', 'User-Agent', 'Organization', 'X-Original-To', 'Delivered-To', 'X-OTRS-Loop', 'X-OTRS-Info', 'X-OTRS-Priority', 'X-OTRS-Queue', 'X-OTRS-Ignore', 'X-OTRS-State', 'X-OTRS-CustomerNo', 'X-OTRS-CustomerUser', 'X-OTRS-ArticleKey1', 'X-OTRS-ArticleKey2', 'X-OTRS-ArticleKey3', 'X-OTRS-ArticleValue1', 'X-OTRS-ArticleValue2', 'X-OTRS-ArticleValue3', 'X-OTRS-SenderType', 'X-OTRS-ArticleType', 'X-OTRS-TicketKey1', 'X-OTRS-TicketKey2', 'X-OTRS-TicketKey3', 'X-OTRS-TicketKey4', 'X-OTRS-TicketKey5', 'X-OTRS-TicketKey6', 'X-OTRS-TicketKey7', 'X-OTRS-TicketKey8', 'X-OTRS-TicketValue1', 'X-OTRS-TicketValue2', 'X-OTRS-TicketValue3', 'X-OTRS-TicketValue4', 'X-OTRS-TicketValue5', 'X-OTRS-TicketValue6', 'X-OTRS-TicketValue7', 'X-OTRS-TicketValue8']; </pre>

B.2.2.15. PostMaster::PreFilterModule###1-Match

Description	Value
Description:	Module to filter and manipulate incoming messages. Block/ignore all spam email with From: noreply@ address.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'PostMaster::PreFilterModule'}->{'1-Match'} = { 'Match' => { }, 'Set' => { 'X-OTRS-Ignore' => 'yes' }, 'Module' => 'Kernel::System::PostMaster::Filter::Match' };</pre>

B.2.2.16. PostMaster::PreFilterModule###2-Match

Description	Value
Description:	Module to filter and manipulate incoming messages. Get a 4 digit number to ticket free text, use regex in Match e. g. From => '(.+?)@.+?', and use () as [***] in Set =>.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'PostMaster::PreFilterModule'}->{'2-Match'} = { 'Match' => { 'Subject' => 'SomeNumber:(\\d\\d\\d\\d\\d)' }, 'Set' => { 'X-OTRS-TicketKey-1' => 'SomeNumber', 'X-OTRS-TicketValue-1' => '[***]' }, 'Module' => 'Kernel::System::PostMaster::Filter::Match' };</pre>

B.2.2.17. PostMaster::PreFilterModule###5-SpamAssassin

Description	Value
Description:	SpamAssassin example setup. Move marked mails to spam queue.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'PostMaster::PreFilterModule'}->{'5-SpamAssassin'} = { 'Set' => { 'X-OTRS-Queue' => 'spam' }, 'CMD' => '/usr/bin/spamassassin grep -i "X-Spam-Status: yes"', 'Module' => 'Kernel::System::PostMaster::Filter::CMD' };</pre>

B.2.2.18. PostMaster::PreFilterModule###000-MatchDBSource

Description	Value
Description:	Module to use database filter storage (use it at first).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'PostMaster::PreFilterModule'}->{'000-MatchDBSource'} = { 'Module' => 'Kernel::System::PostMaster::Filter::MatchDBSource' };</pre>

B.2.2.19. PostMaster::PreFilterModule###999-AgentInterface

Description	Value
Description:	A email agent interface.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'PostMaster::PreFilterModule'}->{'999-AgentInterface'} = { 'Module' => 'Kernel::System::PostMaster::Filter::AgentInterface', 'AgentInterfaceAddress' => 'otrs-agent@example.org' };</pre>

B.2.2.20. SendNoAutoResponseRegExp

Description	Value
Description:	If this regex matches, no message will be send by the autoresponder.

Description	Value
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SendNoAutoResponseRegExp'} = ' (MAILER-DAEMON postmaster abuse</code>

B.2.3. Core::Stats

B.2.3.1. SystemStatsMap###Ticket::Stats1

Description	Value
Description:	Module to print stats about new tickets. Possible values: Print, CSV, GraphLine, GraphBars, GraphPie.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'SystemStatsMap'}->{'Ticket::Stats1'} = { 'UseResultCache' => '0', 'Output' => ['Print', 'CSV', 'Graph'], 'Module' => 'Kernel::System::Stats::NewTickets', 'Desc' => 'New created tickets for each queue in selected month', 'SumRow' => '1', 'SumCol' => '1', 'OutputDefault' => 'Print', 'Name' => 'New Tickets' };</code>

B.2.3.2. SystemStatsMap###Ticket::Stats2

Description	Value
Description:	Module to generate a overview of tickets in a queue at the end of a month. Possible output options are Print, CSV, GraphLine, GraphBars and GraphPie.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1

Description	Value
Required:	0
Config-Setting:	<pre> \$Self->{'SystemStatsMap'}->{'Ticket::Stats2'} = { 'UseResultCache' => '1', 'Output' => ['Print', 'CSV', 'GraphBars'], 'Module' => 'Kernel::System::Stats::TicketOverview', 'Desc' => 'Overview of the tickets in queue at the end of this month', 'SumRow' => '1', 'SumCol' => '1', 'OutputDefault' => 'Print', 'Name' => 'Ticket Overview' }; </pre>

B.2.3.3. SystemStatsMap###Ticket::Stats4

Description	Value
Description:	Module to trace system activities (Replacement of old bin/mkStats.pl). Possible output options are Print, CSV, GraphLine, GraphBars and GraphPie.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'SystemStatsMap'}->{'Ticket::Stats4'} = { 'UseResultCache' => '0', 'Output' => ['Print', 'CSV', 'Graph'], 'Module' => 'Kernel::System::Stats::StateAction', 'Desc' => 'Trace system activities (Replacement of old bin/mkStats.pl', 'SumRow' => '1', 'SumCol' => '1', 'OutputDefault' => 'Graph', 'Name' => 'State Action Overview' }; </pre>

B.2.3.4. SystemStatsMap###Ticket::Stats5

Description	Value
Description:	Module to create a list about accounted time per customer. Possible values for the output are Print and CSV.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'SystemStatsMap'}->{'Ticket::Stats5'} = { 'UseResultCache' => '0', 'Output' => ['Print', 'CSV'], 'Module' => 'Kernel::System::Stats::AccountedTime', 'Desc' => 'A list about accounted time per customer', 'OutputDefault' => 'Print', 'Name' => 'Time Accounting' };</pre>

B.2.4. Core::Ticket

B.2.4.1. Ticket::Hook

Description	Value
Description:	The identifier for a ticket, e.g. Ticket#, Call#, MyTicket#. The default is Ticket#.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Hook'} = 'Ticket#';</pre>

B.2.4.2. Ticket::HookDivider

Description	Value
Description:	The divider between TicketHook and ticket number. E.g ': '.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::HookDivider'} = '';</pre>

B.2.4.3. Ticket::SubjectSize

Description	Value
Description:	Max size of the subjects in an email reply.
Group:	Ticket

Description	Value
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::SubjectSize'} = '60';</code>

B.2.4.4. Ticket::SubjectRe

Description	Value
Description:	The text at the beginning of the subject in a email reply, e.g. RE, AW or AS.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::SubjectRe'} = 'Re';</code>

B.2.4.5. Ticket::CustomQueue

Description	Value
Description:	Name of custom queue. The custom queue is a queue selection of your preferred queues and can be selected in the preferences settings.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::CustomQueue'} = 'My Queues';</code>

B.2.4.6. Ticket::ForceNewStateAfterLock

Description	Value
Description:	Force a new ticket state after lock action. For key you have to define the current state and for value you have to define the next state after lock action.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::ForceNewStateAfterLock'} = { 'new' => 'open' };</code>

B.2.4.7. Ticket::ForceUnlockAfterMove

Description	Value
Description:	Would you like to force tickets to be unlocked after move action?
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::ForceUnlockAfterMove'} = '0';</code>

B.2.4.8. Ticket::ChangeOwnerToEveryone

Description	Value
Description:	Would you like to make it possible to change owner of foreign tickets (useful for ASP).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::ChangeOwnerToEveryone'} = '0';</code>

B.2.4.9. Ticket::NewMessageMode

Description	Value
Description:	Mode how the new message count in the agent navigation bar is used.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::NewMessageMode'} = 'ArticleLastSender';</code>

B.2.4.10. Ticket::NumberGenerator

Description	Value
-------------	-------

Description	Value
Description:	Select the ticket number generator module. "AutoIncrement" just increments the ticket number, the SystemID and the counter are used. The format is SystemID.counter, e.g. 1010138, 1010139. With "Date" the ticket numbers will be generated by the current date, the SystemID and the counter. The format looks like Year.Month.Day.SystemID.counter, e.g. 200206231010138, 200206231010139. With "DateChecksum" the counter will be appended as check summ to the string of date and SystemID. The check summ will be rotated daily. The format looks like Year.Month.Day.SystemID.Counter.CheckSum, e.g. 2002070110101520, 2002070110101535. "Random" generates randomized ticket numbers in the format "SystemID.Random", e.g. 100057866352, 103745394596.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::NumberGenerator'} = 'Kernel::System::Ticket::Number::</code>

B.2.4.11. Ticket::NumberGenerator::MinCounterSize

Description	Value
Description:	If "AutoIncrement" was selected for TicketNumberGenerator, you can set the minimal ticket counter size. Default is 5, this means the counter starts from 10000.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::NumberGenerator::MinCounterSize'} = '5';</code>

B.2.4.12. Ticket::CounterLog

Description	Value
Description:	Log file for the ticket counter.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::CounterLog'} = '<OTRS_CONFIG_Home>/var/log/TicketCounterLog'</code>

B.2.4.13. Ticket::IndexModule

Description	Value
Description:	IndexAccelerator, choose your backend TicketViewAccelerator module. "RuntimeDB" generates each queue view on the fly from ticket table. You won't have performance problems till 60.000 tickets and 6.000 open tickets in your system. "StaticDB" is the most powerfull module. It should be used, if more then 80.000 and 6.000 opten tickets are stored in the system. The module uses a extra ticket_index table, that works like a view. Use the script "bin/RebuildTicketIndex.pl" for initial index update.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::IndexModule'} = 'Kernel::System::Ticket::IndexAccelerator'</code>

B.2.4.14. Ticket::StorageModule

Description	Value
Description:	Select your TicketStorageModule to save the attachments of articles. "DB" stores all data in the database. Don't use this module if big attachments will be stored. "FS" stores the data in the file system, it is faster but webserver user should be the otrs user. You can switch between the modules even with a running system, all data will be availabel and are searched for example.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::StorageModule'} = 'Kernel::System::Ticket::ArticleStorageModule'</code>

B.2.4.15. ArticleDir

Description	Value
Description:	If you have selected "FS" for TicketStorageModule, you have to specify the directory where to store the data.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'ArticleDir'} = '<OTRS_CONFIG_Home>/var/article';</code>

B.2.4.16. Ticket::CustomModule

Description	Value
Description:	A module with custom functions to redefine Kernel::System::Ticket functions (e. g. Kernel::System::Ticket::Custom).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::CustomModule'} = '';</code>

B.2.4.17. Ticket::AgentCanBeCustomer

Description	Value
Description:	Can a agent also be a customer via the agent interface?
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::AgentCanBeCustomer'} = '0';</code>

B.2.4.18. Ticket::ViewableSenderTypes

Description	Value
Description:	Default viewable sender types (default: customer).
Group:	Ticket

Description	Value
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::ViewableSenderTypes'} = ['\customer\''];</code>

B.2.4.19. Ticket::ViewableLocks

Description	Value
Description:	Viewable locks. Default: unlock, tmp_lock.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::ViewableLocks'} = ['\unlock\'' , '\tmp_lock\''];</code>

B.2.4.20. Ticket::ViewableStateType

Description	Value
Description:	State types for a ticket to display.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::ViewableStateType'} = ['new', 'open', 'pending reminder', 'pending auto'];</code>

B.2.4.21. Ticket::UnlockStateType

Description	Value
Description:	The states for unlocked tickets. You can unlock tickets with the script "bin/UnlockTickets.pl".
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::UnlockStateType'} = ['new', 'open'];</code>

B.2.4.22. Ticket::PendingReminderStateType

Description	Value
Description:	State type f?r reminder for pending tickets.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::PendingReminderStateType'} = ['pending reminder'];</code>

B.2.4.23. Ticket::PendingAutoStateType

Description	Value
Description:	Possible states for pending tickets that changed state after reached time limit.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::PendingAutoStateType'} = ['pending auto'];</code>

B.2.4.24. Ticket::StateAfterPending

Description	Value
Description:	States after pending time has reached.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::StateAfterPending'} = { 'pending auto close-' => 'closed unsuccessful', 'pending auto close+' => 'closed successful' };</code>

B.2.4.25. System::Permission

Description	Value
Description:	The different system permissions of the agents in the system (also possible in addition is note, close, pending, customer, freetext, forward and bounce - rw should be the latest in the line).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'System::Permission'} = ['move_into', 'create', 'owner', 'priority', 'rw'];</pre>

B.2.4.26. Ticket::Permission###1-OwnerCheck

Description	Value
Description:	Module to check the owner.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Permission'}->{'1-OwnerCheck'} = { 'Granted' => '0', 'Module' => 'Kernel::System::Ticket::Permission::OwnerCheck', 'Required' => '0' };</pre>

B.2.4.27. Ticket::Permission###2-GroupCheck

Description	Value
Description:	Module to check, if a user is in a special group. Access is granted, if the user is in the specified group and has ro and rw permissions.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Permission'}->{'2-GroupCheck'} = { 'Granted' => '0', 'Module' => 'Kernel::System::Ticket::Permission::GroupCheck', 'Required' => '0' };</pre>

B.2.4.28. CustomerTicket::Permission###1-CustomerUserIDCheck

Description	Value
Description:	Grant access, if customer ID is the same and group is accessible.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerTicket::Permission'}->{'1-CustomerUserIDCheck'} = { 'Granted' => '1', 'Module' => 'Kernel::System::Ticket::CustomerPermission::CustomerUserIDCheck', 'Required' => '0' };</pre>

B.2.4.29. CustomerTicket::Permission###2-CustomerIDCheck

Description	Value
Description:	Module to check customer permissions.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerTicket::Permission'}->{'2-CustomerIDCheck'} = { 'Granted' => '0', 'Module' => 'Kernel::System::Ticket::CustomerPermission::CustomerIDCheck', 'Required' => '1' };</pre>

B.2.4.30. CustomerTicket::Permission###3-GroupCheck

Description	Value
Description:	Module to check the grouppermissions for the access to customer tickets.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerTicket::Permission'}->{'3-GroupCheck'} = { 'Granted' => '0', 'Module' => 'Kernel::System::Ticket::CustomerPermission::GroupCheck', 'Required' => '1' };</pre>

B.2.4.31. CustomerNotifyJustToRealCustomer

Description	Value
Description:	Send customer notifications just to mapped customer. Normally if no customer is mapped, the latest customer sender gets the notification.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerNotifyJustToRealCustomer'} = '0';</code>

B.2.4.32. TicketACL::Default::Action

Description	Value
Description:	Default Ticket Action ACL.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketACL::Default::Action'} = {};</code>

B.2.5. Core::TicketFreeText**B.2.5.1. TicketFreeKey1**

Description	Value
Description:	Define the free key field 1 for tickets. With this setting you can define an new ticket property.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey1'} = { 'Product' => 'Product' };</code>

B.2.5.2. TicketFreeKey1::DefaultSelection

Description	Value
-------------	-------

Description	Value
Description:	The default selection of free text field 1 if more then one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey1::DefaultSelection'} = '';</code>

B.2.5.3. TicketFreeText1

Description	Value
Description:	Define the free text field 1 for tickets. It's a new ticket property.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText1'} = { 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

B.2.5.4. TicketFreeText1::DefaultSelection

Description	Value
Description:	The default selection of free text field 1 if more then one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText1::DefaultSelection'} = 'Notebook';</code>

B.2.5.5. TicketFreeKey2

Description	Value
Description:	Define the free key field 2 for tickets. It's a new ticket property.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0

Description	Value
Required:	0
Config-Setting:	\$Self->{'TicketFreeKey2'} = { " => ' - ' , 'Product' => 'Product' };;

B.2.5.6. TicketFreeKey2::DefaultSelection

Description	Value
Description:	The default selection of free key field 2 if more then one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey2::DefaultSelection'} = "";</code>

B.2.5.7. TicketFreeText2

Description	Value
Description:	Define the free text field 2 for tickets. It's a new ticket poperty.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText2'} = { 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

B.2.5.8. TicketFreeText2::DefaultSelection

Description	Value
Description:	The default selection of free text field 2 if more then one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText2::DefaultSelection'} = 'Notebook';</code>

B.2.5.9. TicketFreeKey3

Description	Value
Description:	Define the free key field 3 for tickets. It's a new ticket property.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey3'} = { 'Product' => 'Product' };</pre>

B.2.5.10. TicketFreeKey3::DefaultSelection

Description	Value
Description:	The default selection of free text field 3 if more then one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey3::DefaultSelection'} = '';</pre>

B.2.5.11. TicketFreeText3

Description	Value
Description:	Define the free text field 3 for tickets. It's a new ticket property.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText3'} = { 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

B.2.5.12. TicketFreeText3::DefaultSelection

Description	Value
Description:	The default selection of free text field 3 if more then one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText

Description	Value
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText3::DefaultSelection'} = 'Notebook';</code>

B.2.5.13. TicketFreeKey4

Description	Value
Description:	Define the free key field 4 for tickets. It's a new ticket property.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey4'} = { 'Product' => 'Product' };</pre>

B.2.5.14. TicketFreeKey4::DefaultSelection

Description	Value
Description:	The default selection of free key field 4 if more then one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey4::DefaultSelection'} = "";</code>

B.2.5.15. TicketFreeText4

Description	Value
Description:	Define the free text field 4 for tickets. It's a new ticket property.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText4'} = { 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

B.2.5.16. TicketFreeText4::DefaultSelection

Description	Value
Description:	The default selection of free text field 4 if more then one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText4::DefaultSelection'} = 'Notebook';</code>

B.2.5.17. TicketFreeKey5

Description	Value
Description:	Define the free key field 5 for tickets. It's a new ticket property.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey5'} = { 'Product' => 'Product' };</code>

B.2.5.18. TicketFreeKey5::DefaultSelection

Description	Value
Description:	The default selection of free key field 5 if more then one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey5::DefaultSelection'} = '';</code>

B.2.5.19. TicketFreeText5

Description	Value
Description:	Define the free text field 5 for tickets. It's a new ticket property.
Group:	Ticket
SubGroup:	Core::TicketFreeText

Description	Value
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText5'} = { 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

B.2.5.20. TicketFreeText5::DefaultSelection

Description	Value
Description:	The default selection of free text field 5 if more then one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText5::DefaultSelection'} = 'Notebook';</pre>

B.2.5.21. TicketFreeKey6

Description	Value
Description:	Define the free key field 6 for tickets. It's a new ticket property.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey6'} = { 'Product' => 'Product' };</pre>

B.2.5.22. TicketFreeKey6::DefaultSelection

Description	Value
Description:	The default selection of free key field 6 if more then one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey6::DefaultSelection'} = '';</pre>

B.2.5.23. TicketFreeText6

Description	Value
Description:	Define the free text field 6 for tickets. It's a new ticket property.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText6'} = { 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

B.2.5.24. TicketFreeText6::DefaultSelection

Description	Value
Description:	The default selection of free text field 6 if more then one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText6::DefaultSelection'} = 'Notebook';</pre>

B.2.5.25. TicketFreeKey7

Description	Value
Description:	Define the free key field 7 for tickets. It's a new ticket property.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey7'} = { 'Product' => 'Product' };</pre>

B.2.5.26. TicketFreeKey7::DefaultSelection

Description	Value
Description:	The default selection of free key field 7 if more then one option is defined.

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey7::DefaultSelection'} = '';</code>

B.2.5.27. TicketFreeText7

Description	Value
Description:	Define the free text field 7 for tickets. It's a new ticket property.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText7'} = { 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

B.2.5.28. TicketFreeText7::DefaultSelection

Description	Value
Description:	The default selection of free text field 7 if more then one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText7::DefaultSelection'} = 'Notebook';</code>

B.2.5.29. TicketFreeKey8

Description	Value
Description:	Define the free key field 8 for tickets. It's a new ticket property.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'TicketFreeKey8'} = { 'Product' => 'Product' };;</pre>

B.2.5.30. TicketFreeKey8::DefaultSelection

Description	Value
Description:	The default selection of free key field 8 if more then one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey8::DefaultSelection'} = "";</code>

B.2.5.31. TicketFreeText8

Description	Value
Description:	Define the free text field 8 for tickets. It's a new ticket property.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText8'} = { 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

B.2.5.32. TicketFreeText8::DefaultSelection

Description	Value
Description:	The default selection of free text field 8 if more then one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText8::DefaultSelection'} = 'Notebook';</code>

B.2.6. Frontend::Admin::ModuleRegistration

B.2.6.1. Frontend::Module###AdminQueue

Description	Value
Description:	Frontend module registration for the AdminQueue object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminQueue'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Queue', 'NavBarModule' => { 'Prio' => '100', 'Block' => 'Block2', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Queue' }, 'Group' => ['admin'] }; </pre>

B.2.6.2. Frontend::Module###AdminResponse

Description	Value
Description:	Frontend module registration for the AdminResponse object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminResponse'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Response', 'NavBarModule' => { 'Prio' => '200', 'Block' => 'Block2', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Responses' }, 'Group' => ['admin'] }; </pre>

B.2.6.3. Frontend::Module###AdminQueueResponses

Description	Value
Description:	Frontend module registration for the AdminQueueResponses object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminQueueResponses'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Responses <-> Queue', 'NavBarModule' => { 'Prio' => '300', 'Block' => 'Block2', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Responses <-> Queue' }, 'Group' => ['admin']; </pre>

B.2.6.4. Frontend::Module###AdminAutoResponse

Description	Value
Description:	Frontend module registration for the AdminAutoResponses object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminAutoResponse'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Auto Responses', 'NavBarModule' => { 'Prio' => '400', 'Block' => 'Block2', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Auto Responses' }, 'Group' => ['admin']; </pre>

B.2.6.5. Frontend::Module###AdminQueueAutoResponse

Description	Value
Description:	Frontend module registration for the AdminQueueAutoResponses object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminQueueAutoResponse'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Auto Responses <-> Queue', 'NavBarModule' => { 'Prio' => '500', 'Block' => 'Block2', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Auto Responses <-> Queue' }, 'Group' => [] }; </pre>

B.2.6.6. Frontend::Module###AdminAttachment

Description	Value
Description:	Frontend module registration for the AdminAttachment object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminAttachment'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Attachment', 'NavBarModule' => { 'Prio' => '600', 'Block' => 'Block2', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Attachments' }, 'Group' => [] }; </pre>

B.2.6.7. Frontend::Module###AdminResponseAttachment

Description	Value
Description:	Frontend module registration for the AdminResponseAttachment object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminResponseAttachment'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Attachments <-> Responses', 'NavBarModule' => { 'Prio' => '700', 'Block' => 'Block2', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Attachments <-> Responses' }, 'Group' => ['admin']; </pre>

B.2.6.8. Frontend::Module###AdminSalutation

Description	Value
Description:	Frontend module registration for the AdminSalutation object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminSalutation'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Salutation', 'NavBarModule' => { 'Prio' => '100', 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Salutations' }, 'Group' => ['admin']; </pre>

B.2.6.9. Frontend::Module###AdminSignature

Description	Value
Description:	Frontend module registration for the AdminSignature object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminSignature'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Signature', 'NavBarModule' => { 'Prio' => '200', 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Signatures' }, 'Group' => ['admin'] }; </pre>

B.2.6.10. Frontend::Module###AdminSystemAddress

Description	Value
Description:	Frontend module registration for the AdminSystemAddress object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminSystemAddress'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'System address', 'NavBarModule' => { 'Prio' => '300', 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Email Addresses' }, 'Group' => ['admin'] }; </pre>

B.2.6.11. Frontend::Module###AdminNotification

Description	Value
Description:	Frontend module registration for the AdminNotification object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminNotification'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'Notification', 'NavBarModule' => { 'Prio' => '400', 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Notifications' }, 'Group' => ['admin'] }; </pre>

B.2.6.12. Frontend::Module###AdminState

Description	Value
Description:	Frontend module registration for the AdminState object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminState'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'State', 'NavBarModule' => { 'Prio' => '700', 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Status' }, 'Group' => ['admin'] }; </pre>

B.2.6.13. Frontend::Module###AdminGenericAgent

Description	Value
-------------	-------

Description	Value
Description:	Frontend module registration for the AdminGenericAgent object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminGenericAgent'} = { 'NavBarName' => 'Admin', 'Description' => 'Admin', 'Title' => 'GenericAgent', 'NavBarModule' => { 'Prio' => '300', 'Block' => 'Block4', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'GenericAgent' }, 'Group' => ['admin'] }; </pre>

B.2.7. Frontend::Agent

B.2.7.1. Ticket::Frontend::PendingDiffTime

Description	Value
Description:	The additional time in sec. which "pending date" shows per default (default: 86400 = 1 day).
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::PendingDiffTime'} = '86400'; </pre>

B.2.7.2. Ticket::Frontend::QueueListType

Description	Value
Description:	Show queues in system as tree or as list.
Group:	Ticket
SubGroup:	Frontend::Agent

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::QueueListType'} = 'tree';</code>

B.2.7.3. Ticket::Frontend::StdResponsesMode

Description	Value
Description:	The standard responses in the agent interface can be displayed as form or links.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::StdResponsesMode'} = 'Link';</code>

B.2.7.4. Ticket::Frontend::TextAreaEmail

Description	Value
Description:	Width of compose email windows.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::TextAreaEmail'} = '78';</code>

B.2.7.5. Ticket::Frontend::TextAreaNote

Description	Value
Description:	Width of compose note windows.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::TextAreaNote'} = '70';</code>

B.2.7.6. Ticket::Frontend::CustomerInfoCompose

Description	Value
Description:	Show customer user info in Compose screen (Phone and Email)?
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerInfoCompose'} = '1';</code>

B.2.7.7. Ticket::Frontend::CustomerInfoComposeMaxSize

Description	Value
Description:	Max size (in characters) of customer info table - Compose screen (Phone and Email).
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerInfoComposeMaxSize'} = '22';</code>

B.2.7.8. Ticket::Frontend::CustomerInfoZoom

Description	Value
Description:	Show customer user info in ticket zoom view?
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerInfoZoom'} = '1';</code>

B.2.7.9. Ticket::Frontend::CustomerInfoZoomMaxSize

Description	Value
-------------	-------

Description	Value
Description:	Max size (in characters) of customer info table - Zoom.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerInfoZoomMaxSize'} = '22';</code>

B.2.7.10. Ticket::Frontend::CustomerInfoQueue

Description	Value
Description:	Show customer user info in Queue view?
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerInfoQueue'} = '0';</code>

B.2.7.11. Ticket::Frontend::CustomerInfoQueueMaxSize

Description	Value
Description:	Max size (in characters) of customer info table - Queue.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerInfoQueueMaxSize'} = '18';</code>

B.2.7.12. Ticket::Frontend::AccountTime

Description	Value
Description:	Aktivate time accounting.
Group:	Ticket
SubGroup:	Frontend::Agent

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AccountTime'} = '1';</code>

B.2.7.13. Ticket::Frontend::TimeUnits

Description	Value
Description:	Your choice of your used time units (e.g. work units, hours, minutes).
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::TimeUnits'} = '(work units)';</code>

B.2.7.14. Ticket::Frontend::NeedAccountedTime

Description	Value
Description:	Does the time have to be accounted?
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NeedAccountedTime'} = '0';</code>

B.2.7.15. Ticket::Frontend::NeedSpellCheck

Description	Value
Description:	Do Newly composed messages have to be spell checked?
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NeedSpellCheck'} = '0';</code>

B.2.7.16. Ticket::Frontend::BulkFeature

Description	Value
Description:	Do you want to activate a agent frontend feature to work on more then one ticket at the same time?
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::BulkFeature'} = '1';</code>

B.2.7.17. Ticket::Frontend::BulkFeatureJavaScriptAlert

Description	Value
Description:	Show java script popup if a ticket is selected for bulk action?
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::BulkFeatureJavaScriptAlert'} = '1';</code>

B.2.7.18. Ticket::Frontend::NewOwnerSelection

Description	Value
Description:	Show owner selection in phone and email ticket?
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NewOwnerSelection'} = '1';</code>

B.2.7.19. Ticket::Frontend::NewQueueSelectionType

Description	Value
Description:	To: selection type. "Queue" shows all queues, "SystemAddress" displays all system addresses.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NewQueueSelectionType'} = 'Queue';</code>

B.2.7.20. Ticket::Frontend::NewQueueSelectionString

Description	Value
Description:	If "Queue" was selected for NewQueueSelectionType, you can specify a selection string.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NewQueueSelectionString'} = '<Queue>';</code>

B.2.7.21. Ticket::Frontend::NewQueueOwnSelection

Description	Value
Description:	If this is in use, "just this selection is valid" for the PhoneView.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NewQueueOwnSelection'} = { '2' => 'Second Queue!', '1' => 'First Queue!' };</code>

B.2.7.22. CustomerDBLink

Description	Value
Description:	External link to customer db. E.g. 'http://yourhost/customer.php?CID=\$Data{"CustomerID"}' or "

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerDBLink'} = ' \$Env{"CGIHandle"}?Action=AgentTicketCusto</code>

B.2.7.23. CustomerDBLinkTarget

Description	Value
Description:	Target attribute in Link to external customer DB. E.g. 'target="cdb"'.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerDBLinkTarget'} = "";</code>

B.2.7.24. Frontend::CommonObject###QueueObject

Description	Value
Description:	Path to the file where all settings for the QueueObject object for the frontend are stored.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::CommonObject'}->{'QueueObject'} = 'Kernel::System::</code>

B.2.7.25. Frontend::CommonObject###TicketObject

Description	Value
Description:	Path to the file where all settings for the TicketObject object are stored.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::CommonObject'}->{'TicketObject'} = 'Kernel::System:'</code>

B.2.7.26. Frontend::CommonParam###Action

Description	Value
Description:	Default value for the Action parameter. The Action parameter is used in the scripts of the system.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::CommonParam'}->{'Action'} = 'AgentTicketQueue';</code>

B.2.7.27. Frontend::CommonParam###QueueID

Description	Value
Description:	Default ID for queue, that is used by the system.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::CommonParam'}->{'QueueID'} = '0';</code>

B.2.7.28. Frontend::CommonParam###TicketID

Description	Value
Description:	Default ID for ticket, that is used by the system.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::CommonParam'}->{'TicketID'} = '';</code>

B.2.8. Frontend::Agent::ModuleNotify

B.2.8.1. Frontend::NotifyModule###3-Ticket::AgentTicketSeen

Description	Value
Description:	Modul for the agent interface that to sets tickets to "seen".
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Frontend::NotifyModule'}->{'3-Ticket::AgentTicketSeen'} = { 'Module' => 'Kernel::Output::HTML::NotificationAgentTicketSeen' };</pre>

B.2.8.2. Frontend::NotifyModule###4-Ticket::TicketNotify

Description	Value
Description:	Module to show agents important tickets via the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Frontend::NotifyModule'}->{'4-Ticket::TicketNotify'} = { 'Module' => 'Kernel::Output::HTML::NotificationAgentTicket' };</pre>

B.2.9. Frontend::Agent::ModuleRegistration

B.2.9.1. Frontend::Module###AgentTicketQueue

Description	Value
Description:	Frontend module registration for the AgentTicketQueue object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1

Description	Value
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketQueue'} = { 'NavBar' => ['Prio' => '100', 'Block' => "", 'Image' => 'overview.png', 'NavBar' => 'Ticket', 'Type' => "", 'AccessKey' => 'o', 'Description' => 'Overview of all open Tickets', 'Name' => 'QueueView', 'Link' => 'Action=AgentTicketQueue' }, { 'Prio' => '200', 'Block' => 'ItemArea', 'Image' => 'desktop.png', 'NavBar' => 'Ticket', 'Type' => 'Menu', 'AccessKey' => 't', 'Description' => 'Ticket-Area', 'Name' => 'Ticket', 'Link' => 'Action=AgentTicketQueue' }], 'NavBarName' => 'Ticket', 'Description' => 'Overview of all open Tickets', 'Title' => 'QueueView' }; </pre>

B.2.9.2. Frontend::Module###AgentTicketPhone

Description	Value
Description:	Frontend module registration for the AgentTicketPhone uue object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketPhone'} = { 'NavBar' => [{ 'Prio' => '200', 'Block' => "", 'Image' => 'phone-new.png', 'NavBar' => 'Ticket', 'Type' => "", 'AccessKey' => 'n', 'Description' => 'Create new Phone Ticket', 'Name' => 'Phone-Ticket', 'Link' => 'Action=AgentTicketPhone' }], 'NavBarName' => 'Ticket', 'Description' => 'Create new Phone Ticket', 'Title' => 'Phone-Ticket' }; </pre>

B.2.9.3. Frontend::Module###AgentTicketEmail

Description	Value
Description:	Frontend module registration for the AgentTicketEmail object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketEmail'} = { 'NavBar' => [{ 'Prio' => '210', 'Block' => "", 'Image' => 'mail_new.png', 'NavBar' => 'Ticket', 'Type' => "", 'AccessKey' => 'e', 'Description' => 'Send Email and create a new Ticket', 'Name' => 'Email-Ticket', 'Link' => 'Action=AgentTicketEmail' }], 'NavBarName' => 'Ticket', 'Description' => 'Create new Email Ticket', 'Title' => 'Email-Ticket' }; </pre>

B.2.9.4. Frontend::Module###AgentTicketSearch

Description	Value
-------------	-------

Description	Value
Description:	Frontend module registration for the AgentTicketSearch object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketSearch'} = { 'NavBar' => [{ 'Prio' => '300', 'Block' => "", 'Image' => 'search.png', 'NavBar' => 'Ticket', 'Type' => "", 'AccessKey' => 's', 'Description' => 'Search Tickets', 'Name' => 'Search', 'Link' => 'Action=AgentTicketSearch' }], 'NavBarName' => 'Ticket', 'Description' => 'Search Tickets', 'Title' => 'Search' }; </pre>

B.2.9.5. Frontend::Module###AgentTicketMailbox

Description	Value
Description:	Frontend module registration for the AgentTicketMailbox object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketMailbox'} = { 'NavBarName' => 'Ticket', 'Description' => 'Agent Mailbox', 'Title' => 'Locked Tickets' }; </pre>

B.2.9.6. Frontend::Module###AgentTicketStatusView

Description	Value
Description:	Frontend module registration for the AgentTicketStatusView object in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketStatusView'} = { 'NavBar' => [{ 'Prio' => '110', 'Block' => "", 'Image' => 'overview.png', 'NavBar' => 'Ticket', 'Type' => "", 'AccessKey' => 'v', 'Description' => 'Overview of all open Tickets', 'Name' => 'StatusView', 'Link' => 'Action=AgentTicketStatusView' }], 'NavBarName' => 'Ticket', 'Description' => 'Overview of all open tickets', 'Title' => 'Status View' }; </pre>

B.2.9.7. Frontend::Module###AgentZoom

Description	Value
Description:	Frontend module registration for the AgentZoom object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentZoom'} = { 'NavBarName' => 'Ticket', 'Description' => 'compat module for Ticket Zoom', 'Title' => "" }; </pre>

B.2.9.8. Frontend::Module###AgentTicketZoom

Description	Value
Description:	Frontend module registration for the AgentTicketZoom object in the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketZoom'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket Zoom', 'Title' => 'Zoom' };</pre>

B.2.9.9. Frontend::Module###AgentTicketAttachment

Description	Value
Description:	Frontend module registration for the AgentTicketAttachment object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketAttachment'} = { 'NavBarName' => 'Ticket', 'Description' => 'To download attachments', 'Title' => " };</pre>

B.2.9.10. Frontend::Module###AgentTicketPlain

Description	Value
Description:	Frontend module registration for the AgentTicketPlain object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketPlain'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket plain view of an email', 'Title' => 'Plain' };</pre>

B.2.9.11. Frontend::Module###AgentTicketNote

Description	Value
Description:	Frontend module registration for the AgentTicketNote object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketNote'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket Note', 'Title' => 'Note' };</pre>

B.2.9.12. Frontend::Module###AgentTicketMerge

Description	Value
Description:	Frontend module registration for the AgentTicketNote object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketMerge'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket Merge', 'Title' => 'Merge' };</pre>

B.2.9.13. Frontend::Module###AgentTicketPending

Description	Value
Description:	Frontend module registration for the AgentTicketPending ote object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketPending'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket Pending', 'Title' => 'Pending' };</pre>

B.2.9.14. Frontend::Module###AgentTicketPriority

Description	Value
Description:	Frontend module registration for the AgentTicketPriority object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketPriority'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket Priority', 'Title' => 'Priority' };</pre>

B.2.9.15. Frontend::Module###AgentTicketLock

Description	Value
Description:	Frontend module registration for the AgentTicketLock object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketLock'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket Lock', 'Title' => 'Lock' };</pre>

B.2.9.16. Frontend::Module###AgentTicketMove

Description	Value
Description:	Frontend module registration for the AgentTicketMove object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketMove'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket Move', 'Title' => 'Move' };</pre>

B.2.9.17. Frontend::Module###AgentTicketHistory

Description	Value
Description:	Frontend module registration for the AgentTicketHistory object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketHistory'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket History', 'Title' => 'History' };</pre>

B.2.9.18. Frontend::Module###AgentTicketOwner

Description	Value
Description:	Frontend module registration for the AgentTicketOwner object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketOwner'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket Owner', 'Title' => 'Owner' };</pre>

B.2.9.19. Frontend::Module###AgentTicketCompose

Description	Value
Description:	Frontend module registration for the AgentTicketCompose object in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketCompose'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket Compose Email Answer', 'Title' => 'Compose' };</pre>

B.2.9.20. Frontend::Module###AgentTicketCustomerFollowUp

Description	Value
Description:	Frontend module registration for the AgentTicketCustomerFollowup object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketCustomerFollowUp'} = { 'NavBarName' => 'Ticket', 'Description' => 'Used if a agent can also be a customer', 'Title' => 'Compose Follow up' };</pre>

B.2.9.21. Frontend::Module###AgentTicketBounce

Description	Value
Description:	Frontend module registration for the AgentTicketBounce object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketBounce'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket Compose Bounce Email', 'Title' => 'Bounce' };</pre>

B.2.9.22. Frontend::Module###AgentTicketForward

Description	Value
Description:	Frontend module registration for the AgentTicketForward object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketForward'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket Forward Email', 'Title' => 'Forward' };</pre>

B.2.9.23. Frontend::Module###AgentTicketCustomer

Description	Value
Description:	Frontend module registration for the AgentTicketCustomer object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketCustomer'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket Customer', 'Title' => 'Customer' };</pre>

B.2.9.24. Frontend::Module###AgentTicketClose

Description	Value
Description:	Frontend module registration for the AgentTicketClose object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketClose'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket Close', 'Title' => 'Close' };</pre>

B.2.9.25. Frontend::Module###AgentTicketFreeText

Description	Value
Description:	Frontend module registration for the AgentTicketFreeText object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketFreeText'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket FreeText', 'Title' => 'Free Fields' };</pre>

B.2.9.26. Frontend::Module###AgentTicketPrint

Description	Value
Description:	Frontend module registration for the AgentTicketPrint object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketPrint'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket Print', 'Title' => 'Print' };</pre>

B.2.9.27. Frontend::Module###AgentTicketBulk

Description	Value
Description:	Frontend module registration for the AgentTicketBulk object in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketBulk'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket bulk module', 'Title' => 'Bulk-Action' };</pre>

B.2.10. Frontend::Agent::NavBarModule

B.2.10.1. Frontend::NavBarModule###1-Ticket::LockedTickets

Description	Value
Description:	Agent interface notification module to check the used charset.
Group:	Ticket
SubGroup:	Frontend::Agent::NavBarModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::NavBarModule'}->{'1-Ticket::LockedTickets'} = { 'Module' => 'Kernel::Output::HTML::NavBarLockedTickets' };</pre>

B.2.10.2. Frontend::NavBarModule###2-Ticket::BulkAction

Description	Value
Description:	Module registration for the BulkAction object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::NavBarModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::NavBarModule'}->{'2-Ticket::BulkAction'} = { 'Module' => 'Kernel::Output::HTML::NavBarTicketBulkAction' };</pre>

B.2.11. Frontend::Agent::Preferences

B.2.11.1. PreferencesGroups###NewTicketNotify

Description	Value
Description:	Parameters for the NewTicketNotify object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'NewTicketNotify'} = { 'Prio' => '1000', 'Label' => 'New ticket notification', 'PrefKey' => 'UserSendNewTicketNotification', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' => 'Send me a notification if there is a new ticket in "My Queue"', 'Activ' => '1', 'Data' => { '1' => 'Yes', '0' => 'No' }, 'Colum' => 'Mail Management' }; </pre>

B.2.11.2. PreferencesGroups###FollowUpNotify

Description	Value
Description:	Parameters for the FollowUpNotify object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'FollowUpNotify'} = { 'Prio' => '2000', 'Label' => 'Follow up notification', 'PrefKey' => 'UserSendFollowUpNotification', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' => 'Send me a notification if a customer sends a follow up and I', 'Activ' => '1', 'Data' => { '1' => 'Yes', '0' => 'No' }, 'Colum' => 'Mail Management' }; </pre>

B.2.11.3. PreferencesGroups###LockTimeoutNotify

Description	Value
-------------	-------

Description	Value
Description:	Parameters for the LockTimeoutNotify object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'LockTimeoutNotify'} = { 'Prio' => '3000', 'Label' => 'Ticket lock timeout notification', 'PrefKey' => 'UserSendLockTimeoutNotification', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' => 'Send me a notification if a ticket is unlocked by the system', 'Activ' => '1', 'Data' => { '1' => 'Yes', '0' => 'No' }, 'Colum' => 'Mail Management' }; </pre>

B.2.11.4. PreferencesGroups###MoveNotify

Description	Value
Description:	Parameters for the MoveNotify object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'MoveNotify'} = { 'Prio' => '4000', 'Label' => 'Move notification', 'PrefKey' => 'UserSendMoveNotification', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' => 'Send me a notification if a ticket is moved into one of "My', 'Activ' => '1', 'Data' => { '1' => 'Yes', '0' => 'No' }, 'Colum' => 'Mail Management' }; </pre>

B.2.11.5. PreferencesGroups###CustomQueue

Description	Value
Description:	Parameters for the CustomQueue object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences

Description	Value
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'CustomQueue'} = { 'Prio' => '2000', 'Label' => 'My Queues', 'Module' => 'Kernel::Output::HTML::PreferencesCustomQueue', 'Desc' => 'Your queue selection of your favorite queues. You also get n 'Activ' => '1', 'Colum' => 'Other Options' };</pre>

B.2.11.6. PreferencesGroups###RefreshTime

Description	Value
Description:	Parameters for the RefreshTime object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'RefreshTime'} = { 'Prio' => '3000', 'Label' => 'QueueView refresh time', 'PrefKey' => 'UserRefreshTime', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' => 'Select your QueueView refresh time.', 'Activ' => '1', 'Data' => { " => 'off', '7' => ' 7 minutes', '10' => '10 minutes', '2' => ' 2 minutes', '15' => '15 minutes', '5' => ' 5 minutes' }, 'Colum' => 'Frontend' };</pre>

B.2.11.7. PreferencesGroups###QueueView

Description	Value
Description:	Parameters for the QueueView object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'QueueView'} = { 'Prio' => '3000', 'Label' => 'QueueView', 'DataSelected' => 'AgentTicketQueueTicketView', 'PrefKey' => 'UserQueueView', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' => 'Select your frontend QueueView.', 'Activ' => '1', 'Data' => { 'AgentTicketQueueTicketViewLite' => 'Lite', 'AgentTicketQueueTicketView' => 'Standard' }, 'Colum' => 'Frontend' }; </pre>

B.2.11.8. PreferencesGroups###QueueViewShownTickets

Description	Value
Description:	Parameters for the QueueViewShowTickets object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'QueueViewShownTickets'} = { 'Prio' => '4000', 'Label' => 'Shown Tickets', 'DataSelected' => '15', 'PrefKey' => 'UserQueueViewShowTickets', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' => 'Max. shown Tickets a page in QueueView.', 'Activ' => '1', 'Data' => { '25' => '25', '10' => '10', '20' => '20', '15' => '15' }, 'Colum' => 'Frontend' }; </pre>

B.2.11.9. PreferencesGroups###CreateNextMask

Description	Value
Description:	Parameters for the CreateNextMask object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'CreateNextMask'} = { 'Prio' => '5000', 'Label' => 'Screen after new ticket', 'DataSelected' => "", 'PrefKey' => 'UserCreateNextMask', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' => 'Select your screen after creating a new ticket.', 'Activ' => '1', 'Data' => { " => 'CreateTicket', 'AgentTicketZoom' => 'TicketZoom' }, 'Colum' => 'Frontend' }; </pre>

B.2.12. Frontend::Agent::Ticket::ArticleAttachmentModule

B.2.12.1. Ticket::Frontend::ArticleAttachmentModule###1-Download

Description	Value
Description:	Links in agent zoom for attachments to download.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleAttachmentModule
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::ArticleAttachmentModule'}->{'1-Download'} = 'Module' => 'Kernel::Output::HTML::ArticleAttachmentDownload' }; </pre>

B.2.12.2. Ticket::Frontend::ArticleAttachmentModule###2-HTML-Viewer

Description	Value
Description:	Links in agent zoom for attachments html online viewer.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleAttachmentModule
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::ArticleAttachmentModule'}->{'2-HTML-Viewer'} 'Module' => 'Kernel::Output::HTML::ArticleAttachmentHTMLViewer' }; </pre>

B.2.13. Frontend::Agent::Ticket::ArticleComposeModule

B.2.13.1. Ticket::Frontend::ArticleComposeModule###1-SignEmail

Description	Value
Description:	Module to compose signed messages (PG or S/MIME).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleComposeModule
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ArticleComposeModule'}->{'1-SignEmail'} = { 'Module' => 'Kernel::Output::HTML::ArticleComposeSign' };</pre>

B.2.13.2. Ticket::Frontend::ArticleComposeModule###2-CryptEmail

Description	Value
Description:	Module to crypt composed messages (PGP or s/MIME).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleComposeModule
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ArticleComposeModule'}->{'2-CryptEmail'} = 'Module' => 'Kernel::Output::HTML::ArticleComposeCrypt' };</pre>

B.2.14. Frontend::Agent::Ticket::ArticleViewModule

B.2.14.1. Ticket::Frontend::ArticleViewModule###1-PGP

Description	Value
Description:	Agent interface article notification module to check pgp.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModule
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ArticleViewModule'}->{'1-PGP'} = { 'Module' => 'Kernel::Output::HTML::ArticleCheckPGP' };</pre>

B.2.14.2. Ticket::Frontend::ArticleViewModule###1-SMIME

Description	Value
Description:	A agent interface module to check incoming emails in the Ticket-Zoom-View if the smime-key is available and true.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModule
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ArticleViewModule'}->{'1-SMIME'} = { 'Module' => 'Kernel::Output::HTML::ArticleCheckSMIME' };</pre>

B.2.15. Frontend::Agent::Ticket::ArticleViewModulePre

B.2.15.1. Ticket::Frontend::ArticlePreViewModule###1-PGP

Description	Value
Description:	Agent interface article notification module to check pgp.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModulePre
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ArticlePreViewModule'}->{'1-PGP'} = { 'Module' => 'Kernel::Output::HTML::ArticleCheckPGP' };</pre>

B.2.15.2. Ticket::Frontend::ArticlePreViewModule###1-SMIME

Description	Value
Description:	Agent interface article notification module to check smime.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModulePre
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ArticlePreViewModule'}->{'1-SMIME'} = { 'Module' => 'Kernel::Output::HTML::ArticleCheckSMIME' };</pre>

B.2.16. Frontend::Agent::Ticket::MenuModule

B.2.16.1. Ticket::Frontend::MenuModule###000-Back

Description	Value
Description:	Module to show back link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'000-Back'} = { 'Action' => "", 'Link' => '\$Env{"LastScreenOverview"}&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Back', 'Name' => 'Back' };</pre>

B.2.16.2. Ticket::Frontend::MenuModule###100-Lock

Description	Value
Description:	Module to show lock/unlock link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'100-Lock'} = { 'Action' => 'AgentTicketLock', 'Module' => 'Kernel::Output::HTML::TicketMenuLock', 'Name' => 'Lock' };</pre>

B.2.16.3. Ticket::Frontend::MenuModule###200-History

Description	Value
Description:	Module to show history link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'200-History'} = { 'Action' => 'AgentTicketHistory', 'Link' => 'Action=AgentTicketHistory&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Shows the ticket history!', 'Name' => 'History' };</pre>

B.2.16.4. Ticket::Frontend::MenuModule###210-Print

Description	Value
Description:	Module to show print link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'210-Print'} = { 'Action' => 'AgentTicketPrint', 'LinkParam' => 'target="print"', 'Link' => 'Action=AgentTicketPrint&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Print this ticket!', 'Name' => 'Print' };</pre>

B.2.16.5. Ticket::Frontend::MenuModule###300-Priority

Description	Value
Description:	Module to show priority link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'300-Priority'} = { 'Action' => 'AgentTicketPriority', 'Link' => 'Action=AgentTicketPriority&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Change the ticket priority!', 'Name' => 'Priority' };</pre>

B.2.16.6. Ticket::Frontend::MenuModule###310-FreeText

Description	Value
Description:	Module to show free text link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'310-FreeText'} = { 'Action' => 'AgentTicketFreeText', 'Link' => 'Action=AgentTicketFreeText&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Change the ticket free fields!', 'Name' => 'Free Fields' };</pre>

B.2.16.7. Ticket::Frontend::MenuModule###320-Link

Description	Value
Description:	Module to show link to link a ticket with another object.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'320-Link'} = { 'Action' => 'AgentLinkObject', 'Link' => 'Action=AgentLinkObject&SourceObject=Ticket&SourceID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Link this ticket to an other objects!', 'Name' => 'Link' };</pre>

B.2.16.8. Ticket::Frontend::MenuModule###400-Owner

Description	Value
Description:	Module to show owner link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'400-Owner'} = { 'Action' => 'AgentTicketOwner', 'Link' => 'Action=AgentTicketOwner&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Change the ticket owner!', 'Name' => 'Owner' };</pre>

B.2.16.9. Ticket::Frontend::MenuModule###410-Customer

Description	Value
Description:	Module to show customer link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'410-Customer'} = { 'Action' => 'AgentTicketCustomer', 'Link' => 'Action=AgentTicketCustomer&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Change the ticket customer!', 'Name' => 'Customer' };</pre>

B.2.16.10. Ticket::Frontend::MenuModule###420-Note

Description	Value
Description:	Module to show link to note in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'420-Note'} = { 'Action' => 'AgentTicketNote', 'Link' => 'Action=AgentTicketNote&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Add a note to this ticket!', 'Name' => 'Note' };</pre>

B.2.16.11. Ticket::Frontend::MenuModule###430-Merge

Description	Value
Description:	Module to show merge link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'430-Merge'} = { 'Action' => 'AgentTicketMerge', 'Link' => 'Action=AgentTicketMerge&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Merge this ticket!', 'Name' => 'Merge' };</pre>

B.2.16.12. Ticket::Frontend::MenuModule###440-Pending

Description	Value
Description:	Module to show pending link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'440-Pending'} = { 'Action' => 'AgentTicketPending', 'Link' => 'Action=AgentTicketPending&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Set this ticket to pending!', 'Name' => 'Pending' };</pre>

B.2.16.13. Ticket::Frontend::MenuModule###450-Close

Description	Value
Description:	Module to show close link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'450-Close'} = { 'Action' => 'AgentTicketClose', 'Link' => 'Action=AgentTicketClose&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Close this ticket!', 'Name' => 'Close' };</pre>

B.2.17. Frontend::Agent::Ticket::MenuModulePre

B.2.17.1. Ticket::Frontend::PreMenuModule###100-Lock

Description	Value
Description:	Show lock / unlock link
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::PreMenuModule'}->{'100-Lock'} = { 'Action' => 'AgentTicketLock', 'Module' => 'Kernel::Output::HTML::TicketMenuLock', 'Name' => 'Lock' };</pre>

B.2.17.2. Ticket::Frontend::PreMenuModule###200-Zoom

Description	Value
Description:	Show zoom link
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::PreMenuModule'}->{'200-Zoom'} = { 'Action' => 'AgentTicketZoom', 'Link' => 'Action=AgentTicketZoom&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Look into a ticket!', 'Name' => 'Zoom' }; </pre>

B.2.17.3. Ticket::Frontend::PreMenuModule###210-History

Description	Value
Description:	Show history link
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::PreMenuModule'}->{'210-History'} = { 'Action' => 'AgentTicketHistory', 'Link' => 'Action=AgentTicketHistory&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Shows the ticket history!', 'Name' => 'History' }; </pre>

B.2.17.4. Ticket::Frontend::PreMenuModule###300-Priority

Description	Value
Description:	Show priority link
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::PreMenuModule'}->{'300-Priority'} = { 'Action' => 'AgentTicketPriority', 'Link' => 'Action=AgentTicketPriority&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Change the ticket priority!', 'Name' => 'Priority' };</pre>

B.2.17.5. Ticket::Frontend::PreMenuModule###420-Note

Description	Value
Description:	Show note link
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::PreMenuModule'}->{'420-Note'} = { 'Action' => 'AgentTicketNote', 'Link' => 'Action=AgentTicketNote&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Add a note to this ticket!', 'Name' => 'Note' };</pre>

B.2.17.6. Ticket::Frontend::PreMenuModule###440-Close

Description	Value
Description:	Show close link
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::PreMenuModule'}->{'440-Close'} = { 'Action' => 'AgentTicketClose', 'Link' => 'Action=AgentTicketClose&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Close this ticket!', 'Name' => 'Close' };</pre>

B.2.17.7. Ticket::Frontend::PreMenuModule###450-Delete

Description	Value
Description:	Show delete link
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::PreMenuModule'}->{'450-Delete'} = { 'Action' => 'AgentTicketMove', 'Link' => 'Action=AgentTicketMove&TicketID=\$Data{"TicketID"}&DestQueue= 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Delete this ticket!', 'Name' => 'Delete' }; </pre>

B.2.17.8. Ticket::Frontend::PreMenuModule###460-Spam

Description	Value
Description:	Show spam link
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::PreMenuModule'}->{'460-Spam'} = { 'Action' => 'AgentTicketMove', 'Link' => 'Action=AgentTicketMove&TicketID=\$Data{"TicketID"}&DestQueue= 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Description' => 'Mark as Spam!', 'Name' => 'Spam' }; </pre>

B.2.18. Frontend::Agent::Ticket::ViewBounce**B.2.18.1. Ticket::Frontend::BounceState**

Description	Value
Description:	Default next state after bouncing a ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::BounceState'} = 'closed successful';</code>

B.2.18.2. Ticket::DefaultNextBounceStateType

Description	Value
Description:	Next possible states when a ticket is bounced.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::DefaultNextBounceStateType'} = ['open', 'closed'];</code>

B.2.18.3. Ticket::Frontend::BounceText

Description	Value
Description:	Default note text
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::BounceText'} = 'Your email with ticket numb</code>

B.2.19. Frontend::Agent::Ticket::ViewClose

B.2.19.1. Ticket::Frontend::CloseNoteType

Description	Value
Description:	Default note type for closed tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CloseNoteType'} = 'note-internal';</code>

B.2.19.2. Ticket::Frontend::CloseSubject

Description	Value
Description:	Default text for subject of closed tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CloseSubject'} = '\$Text{"Close"}!';</code>

B.2.19.3. Ticket::Frontend::CloseText

Description	Value
Description:	Default text for closed tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CloseText'} = '';</code>

B.2.19.4. Ticket::Frontend::CloseState

Description	Value
Description:	Default state for closed tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CloseState'} = 'closed successful';</code>

B.2.19.5. Ticket::DefaultCloseNextStateType

Description	Value
Description:	Next possible states for close tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::DefaultCloseNextStateType'} = ['closed'];</code>

B.2.20. Frontend::Agent::Ticket::ViewCompose

B.2.20.1. Ticket::DefaultNextComposeType

Description	Value
Description:	Default next state after a ticket is composed, answered, e.g..
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::DefaultNextComposeType'} = 'open';</code>

B.2.20.2. Ticket::DefaultNextComposeStateType

Description	Value
Description:	Next possible states after composing / answering a ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::DefaultNextComposeStateType'} = ['open', 'closed', 'pending auto', 'pending reminder'];</pre>

B.2.20.3. Ticket::Frontend::ResponseFormat

Description	Value
Description:	Format of responses.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ResponseFormat'} = '\$QData{"Salutation"} \$QData{"OrigFrom"} \$Text{"wrote"}: \$QData{"Body"} \$QData{"StdResponse"} \$QData{"Signature"} ';</pre>

B.2.20.4. Ticket::Frontend::ComposeReplaceSenderAddress

Description	Value
Description:	Replace on compose answer the original sender with current customers email address.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ComposeReplaceSenderAddress'} = '0';</pre>

B.2.21. Frontend::Agent::Ticket::ViewEmailNew

B.2.21.1. Ticket::Frontend::EmailPriority

Description	Value
Description:	Default priority for new email tickets.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::EmailPriority'} = '3 normal';</code>

B.2.21.2. Ticket::Frontend::EmailNewArticleType

Description	Value
Description:	Default article type for new email tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::EmailNewArticleType'} = 'email-external';</code>

B.2.21.3. Ticket::Frontend::EmailNewSenderType

Description	Value
Description:	Default sender type for new email tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::EmailNewSenderType'} = 'agent';</code>

B.2.21.4. Ticket::Frontend::EmailNewHistoryType

Description	Value
Description:	Default history type for new email tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::EmailNewHistoryType'} = 'EmailAgent';</code>

B.2.21.5. Ticket::Frontend::EmailNewHistoryComment

Description	Value
Description:	Default history comment for new email tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::EmailNewHistoryComment'} = '';</code>

B.2.21.6. Ticket::Frontend::EmailNewNote

Description	Value
Description:	Default text for new email tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::EmailNewNote'} = '';</code>

B.2.21.7. Ticket::EmailDefaultNextStateType

Description	Value
Description:	Next possible states after email ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::EmailDefaultNextStateType'} = ['open', 'pending auto', 'pending reminder', 'closed'];</code>

B.2.21.8. Ticket::Frontend::EmailNewNextState

Description	Value
Description:	Default next state after an email ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::EmailNewNextState'} = 'open';</code>

B.2.21.9. Ticket::Frontend::EmailNewShownCustomerTickets

Description	Value
Description:	Max. shown customer history tickets in email-ticket mask.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::EmailNewShownCustomerTickets'} = '10';</code>

B.2.22. Frontend::Agent::Ticket::ViewForward**B.2.22.1. Ticket::DefaultNextForwardStateType**

Description	Value
Description:	Next possible states when forwarding a ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::DefaultNextForwardStateType'} = ['open', 'closed'];</code>

B.2.22.2. Ticket::Frontend::ForwardArticleTypes

Description	Value
Description:	Possible email types.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::ForwardArticleTypes'} = ['email-external', 'email-internal'];</code>

B.2.22.3. Ticket::Frontend::ForwardArticleType

Description	Value
Description:	Default type of forwarded message.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::ForwardArticleType'} = 'email-external';</code>

B.2.23. Frontend::Agent::Ticket::ViewHistory**B.2.23.1. Ticket::Frontend::HistoryOrder**

Description	Value
Description:	Show history order reverse in the agent interface under ticket history.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewHistory
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::HistoryOrder'} = 'normal';</code>

B.2.24. Frontend::Agent::Ticket::ViewMailbox

B.2.24.1. Ticket::Frontend::MailboxSortBy::Default

Description	Value
Description:	Mailbox sort by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::MailboxSortBy::Default'} = 'Age';</code>

B.2.24.2. Ticket::Frontend::MailboxOrder::Default

Description	Value
Description:	Mailbox order default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::MailboxOrder::Default'} = 'Up';</code>

B.2.25. Frontend::Agent::Ticket::ViewMerge

B.2.25.1. Ticket::Frontend::MergeText

Description	Value
Description:	Default merge text
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::MergeText'} = 'Your email with ticket number';</code>

B.2.26. Frontend::Agent::Ticket::ViewMove

B.2.26.1. Ticket::Frontend::MoveType

Description	Value
Description:	The queue selection can be shown as dropdown box or in a new opened window.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::MoveType'} = 'form';</code>

B.2.26.2. Ticket::Frontend::MoveSetState

Description	Value
Description:	Set new ticket state after ticket move?
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::MoveSetState'} = '1';</code>

B.2.26.3. Ticket::DefaultNextMoveStateType

Description	Value
Description:	Possible states after a ticket is moved.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::DefaultNextMoveStateType'} = ['open', 'closed'];</code>

B.2.27. Frontend::Agent::Ticket::ViewNote

B.2.27.1. Ticket::Frontend::NoteSetState

Description	Value
Description:	Would you like to set the state of a ticket if a note is created by an agent?
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NoteSetState'} = '0';</code>

B.2.27.2. Ticket::Frontend::NoteInformInvolvedAgent

Description	Value
Description:	Show selection of involved agents?
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NoteInformInvolvedAgent'} = '0';</code>

B.2.27.3. Ticket::Frontend::NoteInformAgent

Description	Value
Description:	Show selection of agents to inform?
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NoteInformAgent'} = '0';</code>

B.2.27.4. Ticket::DefaultNextNoteStateType

Description	Value
Description:	Default next states after adding a note.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::DefaultNextNoteStateType'} = ['new', 'open', 'closed'];</code>

B.2.27.5. Ticket::Frontend::NoteType

Description	Value
Description:	Default note type.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NoteType'} = 'note-internal';</code>

B.2.27.6. Ticket::Frontend::NoteTypes

Description	Value
Description:	Specify the different note types, that you want to use in your system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NoteTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };</code>

B.2.27.7. Ticket::Frontend::NoteSubject

Description	Value
Description:	Default note subject.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NoteSubject'} = '\$Text{"Note"}!';</code>

B.2.27.8. Ticket::Frontend::NoteText

Description	Value
Description:	Default note text.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NoteText'} = '';</code>

B.2.28. Frontend::Agent::Ticket::ViewOwner

B.2.28.1. Ticket::Frontend::OwnerSetState

Description	Value
Description:	Would you like to set the state of a ticket if a owner will be set in frontend?
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::OwnerSetState'} = '0';</code>

B.2.28.2. Ticket::DefaultNextOwnerStateType

Description	Value
Description:	Default next states after owner selection.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::DefaultNextOwnerStateType'} = ['open', 'closed'];</code>

B.2.28.3. Ticket::Frontend::OwnerSubject

Description	Value
Description:	Default owner update subject.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::OwnerSubject'} = '\$Text{"Owner Update"}';</code>

B.2.29. Frontend::Agent::Ticket::ViewPending

B.2.29.1. Ticket::Frontend::PendingSubject

Description	Value
Description:	Default text in the subject of pending tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PendingSubject'} = '\$Text{"Pending"}!';</code>

B.2.29.2. Ticket::Frontend::PendingText

Description	Value
Description:	Default text for pending tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PendingText'} = '';</code>

B.2.29.3. Ticket::Frontend::PendingState

Description	Value
Description:	Default state of pending ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PendingState'} = 'pending reminder';</code>

B.2.29.4. Ticket::DefaultPendingNextStateType

Description	Value
Description:	Next possible states for pending tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::DefaultPendingNextStateType'} = ['pending reminder', 'pending auto'];</code>

B.2.29.5. Ticket::Frontend::PendingNoteType

Description	Value
Description:	Default note type for pending tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PendingNoteType'} = 'note-internal';</code>

B.2.30. Frontend::Agent::Ticket::ViewPhone

B.2.30.1. Ticket::Frontend::PhoneArticleType

Description	Value
Description:	Default note type for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhone
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PhoneArticleType'} = 'phone';</code>

B.2.30.2. Ticket::Frontend::PhoneSenderType

Description	Value
Description:	Default sender type for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhone
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PhoneSenderType'} = 'agent';</code>

B.2.30.3. Ticket::Frontend::PhoneSubject

Description	Value
Description:	Default subject for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhone
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PhoneSubject'} = '\$Text{"Phone call"}!';</code>

B.2.30.4. Ticket::Frontend::PhoneNote

Description	Value
Description:	Default note text for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhone
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PhoneNote'} = 'Customer called';</code>

B.2.30.5. Ticket::PhoneDefaultNextStateType

Description	Value
Description:	Next possible states after phone note.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhone
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::PhoneDefaultNextStateType'} = ['open', 'pending auto', 'pending reminder', 'closed'];</code>

B.2.30.6. Ticket::Frontend::PhoneNextState

Description	Value
Description:	Default next state after adding a phone note.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhone
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PhoneNextState'} = 'closed successful';</code>

B.2.30.7. Ticket::Frontend::PhoneHistoryType

Description	Value
-------------	-------

Description	Value
Description:	Default history type for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhone
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PhoneHistoryType'} = 'PhoneCallAgent';</code>

B.2.30.8. Ticket::Frontend::PhoneHistoryComment

Description	Value
Description:	Default history comment for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhone
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PhoneHistoryComment'} = '';</code>

B.2.30.9. Ticket::Frontend::PhonePriority

Description	Value
Description:	Default priority for new phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhone
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PhonePriority'} = '3 normal';</code>

B.2.31. Frontend::Agent::Ticket::ViewPhoneNew

B.2.31.1. Ticket::Frontend::PhoneNewArticleType

Description	Value
Description:	Default article type for a new phone ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PhoneNewArticleType'} = 'phone';</code>

B.2.31.2. Ticket::Frontend::PhoneNewSenderType

Description	Value
Description:	Default sender type for new phone ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PhoneNewSenderType'} = 'customer';</code>

B.2.31.3. Ticket::Frontend::PhoneNewSubject

Description	Value
Description:	Default subject for new phone ticket (e.g. 'Phone call').
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PhoneNewSubject'} = '';</code>

B.2.31.4. Ticket::Frontend::PhoneNewNote

Description	Value
Description:	Default text for new phone tickets. E.g 'New ticket via call.'
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PhoneNewNote'} = '';</code>

B.2.31.5. Ticket::Frontend::PhoneNewNextState

Description	Value
Description:	Default next state for a new phone ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PhoneNewNextState'} = 'open';</code>

B.2.31.6. Ticket::Frontend::PhoneNewShownCustomerTickets

Description	Value
Description:	Max. shown customer history tickets in phone-ticket mask.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PhoneNewShownCustomerTickets'} = '10';</code>

B.2.31.7. Ticket::Frontend::PhoneNewHistoryType

Description	Value
Description:	Default history type for new phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PhoneNewHistoryType'} = 'PhoneCallCustomer'</code>

B.2.31.8. Ticket::Frontend::PhoneNewHistoryComment

Description	Value
Description:	Default history comment for new phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PhoneNewHistoryComment'} = '';</code>

B.2.32. Frontend::Agent::Ticket::ViewQueue

B.2.32.1. Ticket::QueueViewAllPossibleTickets

Description	Value
Description:	Show all ro and rw queues in the queue view (not only rw queues)?
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::QueueViewAllPossibleTickets'} = '0';</code>

B.2.32.2. HighlightAge1

Description	Value
Description:	Set the age (first level) for highlighting of old queue (Age in minutes).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'HighlightAge1'} = '1440';</code>

B.2.32.3. HighlightColor1

Description	Value
Description:	Set the color for highlighting of old queue (first level).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'HighlightColor1'} = 'orange';</code>

B.2.32.4. HighlightAge2

Description	Value
Description:	Set the age (second level) for highlighting of old queue (Age in minutes).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'HighlightAge2'} = '2880';</code>

B.2.32.5. HighlightColor2

Description	Value
Description:	Set the color for highlighting of old queue (second level).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'HighlightColor2'} = 'red';</code>

B.2.32.6. Ticket::Frontend::NoEscalationGroup

Description	Value
-------------	-------

Description	Value
Description:	Don't show escalated tickets to agents, who have write permissions for their specified group.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NoEscalationGroup'} = 'some_group';</code>

B.2.32.7. Ticket::Frontend::QueueMaxShown

Description	Value
Description:	Max shown tickets in queue view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::QueueMaxShown'} = '1200';</code>

B.2.32.8. Ticket::Frontend::QueueSort

Description	Value
Description:	Sort the tickets when a single queue is selected in the QueueView in ascending or descending order (after the tickets are sorted by priority). Values: 0 = ascending (oldest on top, default), 1 = descending (youngest on top). Use the queueID for the key and 0 or 1 for value.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::QueueSort'} = { '7' => '1', '3' => '0'};</code>

B.2.32.9. Ticket::Frontend::QueueSortByDefault

Description	Value
-------------	-------

Description	Value
Description:	Default sort criteria for all queues displayed in the QueueView after sort by priority is done
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::QueueSortByDefault'} = 'Age';</code>

B.2.32.10. Ticket::Frontend::QueueOrder::Default

Description	Value
Description:	Sort order for all queues in the QueueView after priority sort.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::QueueOrder::Default'} = 'Up';</code>

B.2.33. Frontend::Agent::Ticket::ViewSearch

B.2.33.1. Ticket::Frontend::SearchLimit

Description	Value
Description:	Search limit for ticket frontend.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::SearchLimit'} = '5000';</code>

B.2.33.2. Ticket::Frontend::SearchPageShown

Description	Value
-------------	-------

Description	Value
Description:	Default of shown article a page.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::SearchPageShown'} = '40';</code>

B.2.33.3. Ticket::Frontend::SearchViewableTicketLines

Description	Value
Description:	Viewable ticket lines by search utility.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::SearchViewableTicketLines'} = '10';</code>

B.2.33.4. Ticket::Frontend::SortBy::Default

Description	Value
Description:	Search result sort by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::SortBy::Default'} = 'Age';</code>

B.2.33.5. Ticket::Frontend::Order::Default

Description	Value
Description:	Search result order default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::Order::Default'} = 'Down';</code>

B.2.33.6. Ticket::Frontend::SearchArticleCSVTree

Description	Value
Description:	Export also whole article tree in search result export (take care of the system performance!).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::SearchArticleCSVTree'} = '0';</code>

B.2.33.7. Ticket::Frontend::SearchCSVData

Description	Value
Description:	Used data to export the search result in CSV format..
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::SearchCSVData'} = ['TicketNumber', 'Age', 'Created', 'State', 'Priority', 'Queue', 'Lock', 'Owner', 'UserFirstname', 'UserLastname', 'CustomerID', 'CustomerName', 'From', 'Subject', 'AccountedTime', 'TicketFreeKey1', 'TicketFreeText1', 'TicketFreeKey2', 'TicketFreeText2', 'TicketFreeKey3', 'TicketFreeText3', 'TicketFreeKey4', 'TicketFreeText4', 'TicketFreeKey5', 'TicketFreeText5', 'TicketFreeKey6', 'TicketFreeText6', 'TicketFreeKey7', 'TicketFreeText7', 'TicketFreeKey8', 'TicketFreeText8', 'TicketFreeTime1', 'TicketFreeTime2', 'CustomerName', 'ArticleTree'];</pre>

B.2.33.8. Ticket::Frontend::Search::DB::DSN

Description	Value
Description:	If you want to use a mirror database for agent ticket fulltext search, specify the DSN to this database.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::Search::DB::DSN'} = 'DBI:mysql:database=mir</pre>

B.2.33.9. Ticket::Frontend::Search::DB::User

Description	Value
Description:	If you want to use a mirror database for agent ticket fulltext search, you can specify the user to authenticate to this database.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::Search::DB::User'} = 'some_user';</code>

B.2.33.10. Ticket::Frontend::Search::DB::Password

Description	Value
Description:	If you want to use a mirror database for agent ticket fulltext search, you can specify the password to authenticate to this database.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::Search::DB::Password'} = 'some_password';</code>

B.2.34. Frontend::Agent::Ticket::ViewStatus

B.2.34.1. Ticket::Frontend::StatusView::ViewableTicketsPage

Description	Value
Description:	Shows all open tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::StatusView::ViewableTicketsPage'} = '50';</code>

B.2.34.2. Ticket::Frontend::StatusSortBy::Default

Description	Value
Description:	Queue sort by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::StatusSortBy::Default'} = 'Age';</code>

B.2.34.3. Ticket::Frontend::StatusOrder::Default

Description	Value
Description:	Queue order default (after priority sort). ASC: oldest on top. DESC: youngest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::StatusOrder::Default'} = 'Down';</code>

B.2.35. Frontend::Agent::Ticket::ViewZoom

B.2.35.1. Ticket::Frontend::ZoomExpand

Description	Value
Description:	Show article expanded in agent interface under ticket zoom.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::ZoomExpand'} = '0';</code>

B.2.35.2. Ticket::Frontend::ZoomExpandSort

Description	Value
Description:	Show article normal or in reverse order in the agent interface under ticket zoom.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::ZoomExpandSort'} = 'normal';</code>

B.2.35.3. Ticket::ZoomAttachmentDisplay

Description	Value
Description:	Shows a icon in the ticket zoom, if a article as an attachment.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::ZoomAttachmentDisplay'} = '1';</code>

B.2.35.4. Ticket::ZoomTimeDisplay

Description	Value
Description:	Displays the time that is accounted for an article in the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::ZoomTimeDisplay'} = '0';</code>

B.2.36. Frontend::Customer

B.2.36.1. CustomerPriority

Description	Value
Description:	Is the customer allowed to set the priority for a ticket?
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPriority'} = '1';</code>

B.2.36.2. CustomerDefaultPriority

Description	Value
Description:	Default priority of new customer tickets.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerDefaultPriority'} = '3 normal';</code>

B.2.36.3. CustomerDefaultState

Description	Value
Description:	Ddefault state of new customer tickets.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerDefaultState'} = 'new';</code>

B.2.36.4. CustomerNextScreenAfterNewTicket

Description	Value
Description:	Next screen after new customer ticket.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerNextScreenAfterNewTicket'} = 'CustomerTicketOverview';</code>

B.2.36.5. CustomerPanelArticleType

Description	Value
Description:	Type of note.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelArticleType'} = 'webrequest';</code>

B.2.36.6. CustomerPanelSenderType

Description	Value
Description:	Sender type for requests from the customer panel.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelSenderType'} = 'customer';</code>

B.2.36.7. CustomerPanelHistoryType

Description	Value
Description:	Default history type in the customer panel.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelHistoryType'} = 'FollowUp';</code>

B.2.36.8. CustomerPanelHistoryComment

Description	Value
Description:	Default history comment in the customer panel.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelHistoryComment'} = '';</code>

B.2.36.9. CustomerPanelDefaultNextComposeType

Description	Value
Description:	Next state for ticket after customer followup.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelDefaultNextComposeType'} = 'open';</code>

B.2.36.10. CustomerPanelNextComposeState

Description	Value
Description:	Next commpose state for customer tickets.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelNextComposeState'} = '1';</code>

B.2.36.11. Ticket::CustomerPanelDefaultNextComposeStateType

Description	Value
Description:	Next possible states for customer tickets.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::CustomerPanelDefaultNextComposeStateType'} = ['open', 'closed'];</code>

B.2.36.12. CustomerPanelNewArticleType

Description	Value
Description:	Default article type.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelNewArticleType'} = 'webrequest';</code>

B.2.36.13. CustomerPanelNewSenderType

Description	Value
Description:	Sender type for new tickets from the customer panel.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelNewSenderType'} = 'customer';</code>

B.2.36.14. CustomerPanelNewHistoryType

Description	Value
Description:	Default history type.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelNewHistoryType'} = 'WebRequestCustomer';</code>

B.2.36.15. CustomerPanelNewHistoryComment

Description	Value
Description:	Comment for new history entrys in the customer panel.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelNewHistoryComment'} = '';</code>

B.2.36.16. CustomerPanelSelectionType

Description	Value
Description:	To: selection type. "Queue" shows all queues, "SystemAddress" displays all system addresses.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelSelectionType'} = 'Queue';</code>

B.2.36.17. CustomerPanelSelectionString

Description	Value
Description:	If "Queue" was selected for CustomerPanelSelectionType, you can specify a selection string (e. g. 'Queue: <Queue> - <QueueComment>').
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'CustomerPanelSelectionString'} = '<Queue>';</code>

B.2.36.18. CustomerPanelOwnSelection

Description	Value
Description:	Here you can insert the Queues, witch the Customer can select in the Customer-Interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'CustomerPanelOwnSelection'} = { 'Misc' => 'Second Queue', 'Junk' => 'First Queue' };</code>

B.2.36.19. CustomerPanel::NewTicketQueueSelectionModule

Description	Value
-------------	-------

Description	Value
Description:	Module for To-selection in new ticket screen in the customer panel.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanel::NewTicketQueueSelectionModule'} = 'Kernel::Out</code>

B.2.36.20. Ticket::CustomerTicketSearch::SearchLimit

Description	Value
Description:	Default limit for ticket search.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::CustomerTicketSearch::SearchLimit'} = '5000';</code>

B.2.36.21. Ticket::CustomerTicketSearch::SearchPageShown

Description	Value
Description:	Default number of articles per page after searching through tickets.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::CustomerTicketSearch::SearchPageShown'} = '40';</code>

B.2.36.22. Ticket::CustomerTicketSearch::SortBy::Default

Description	Value
Description:	Search result sort by default.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::CustomerTicketSearch::SortBy::Default'} = 'Age';</code>

B.2.36.23. Ticket::CustomerTicketSearch::Order::Default

Description	Value
Description:	Search result order default.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::CustomerTicketSearch::Order::Default'} = 'Down';</code>

B.2.36.24. CustomerFrontend::CommonObject###QueueObject

Description	Value
Description:	Path to the file where all settings for the QueueObject object for the customer interface are stored.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerFrontend::CommonObject'}->{'QueueObject'} = 'Kernel::';</code>

B.2.36.25. CustomerFrontend::CommonObject###TicketObject

Description	Value
Description:	Path to the file where all settings for the TicketObject object for the customer interface are stored.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerFrontend::CommonObject'}->{'TicketObject'} = 'Kernel:'</code>

B.2.36.26. CustomerFrontend::CommonParam###Action

Description	Value
Description:	Standardwert für den Action-Parameter innerhalb des Kunden-Interfaces.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerFrontend::CommonParam'}->{'Action'} = 'CustomerTicket'</code>

B.2.36.27. CustomerFrontend::CommonParam###QueueID

Description	Value
Description:	Standard parameter for the QueueID in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerFrontend::CommonParam'}->{'QueueID'} = '0';</code>

B.2.36.28. CustomerFrontend::CommonParam###TicketID

Description	Value
Description:	Standard parameter for the TicketID in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'CustomerFrontend::CommonParam'}->{'TicketID'} = "";</code>

B.2.37. Frontend::Customer::ModuleRegistration

B.2.37.1. CustomerFrontend::Module###CustomerTicketOverView

Description	Value
Description:	Frontend module registration for the CustomerTicketOverview object in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerFrontend::Module'}->{'CustomerTicketOverView'} = { 'NavBar' => [{ 'Prio' => '110', 'Block' => "", 'Image' => 'ticket.png', 'NavBar' => "", 'Type' => "", 'AccessKey' => 'm', 'Description' => 'MyTickets', 'Name' => 'MyTickets', 'Link' => 'Action=CustomerTicketOverView&Type=MyTickets' }, { 'Prio' => '120', 'Block' => "", 'Image' => 'tickets.png', 'NavBar' => "", 'Type' => "", 'AccessKey' => 'c', 'Description' => 'CompanyTickets', 'Name' => 'CompanyTickets', 'Link' => 'Action=CustomerTicketOverView&Type=CompanyTickets' }], 'NavBarName' => 'Ticket', 'Description' => 'Overview of customer tickets', 'Title' => 'Overview' }; </pre>

B.2.37.2. CustomerFrontend::Module###CustomerTicketMessage

Description	Value
Description:	Frontend module registration for the CustomerTicketMessage object in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerFrontend::Module'}->{'CustomerTicketMessage'} = { 'NavBar' => [{ 'Prio' => '100', 'Block' => "", 'Image' => 'new.png', 'NavBar' => "", 'Type' => "", 'AccessKey' => 'n', 'Description' => 'Create new Ticket', 'Name' => 'New Ticket', 'Link' => 'Action=CustomerTicketMessage' }], 'NavBarName' => 'Ticket', 'Description' => 'Create and updated tickets', 'Title' => 'Message' }; </pre>

B.2.37.3. CustomerFrontend::Module###CustomerTicketZoom

Description	Value
Description:	Frontend module registration for the CustomerTicketZoom object in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerFrontend::Module'}->{'CustomerTicketZoom'} = { 'NavBarName' => 'Ticket', 'Description' => 'Ticket zoom view', 'Title' => 'Zoom' }; </pre>

B.2.37.4. CustomerFrontend::Module###CustomerZoom

Description	Value
-------------	-------

Description	Value
Description:	Frontend module registration for the CustomerZoom object in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}->{'CustomerZoom'} = { 'NavBarName' => "", 'Description' => 'compat mod', 'Title' => " };</pre>

B.2.37.5. CustomerFrontend::Module###CustomerTicketAttachment

Description	Value
Description:	Frontend module registration for the CustomerTicketAttachment object in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}->{'CustomerTicketAttachment'} = { 'NavBarName' => "", 'Description' => 'To download attachments', 'Title' => " };</pre>

B.2.37.6. CustomerFrontend::Module###CustomerTicketSearch

Description	Value
Description:	Frontend module registration for the CustomerTicketSearch object in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'CustomerFrontend::Module'}->{'CustomerTicketSearch'} = { 'NavBar' => [{ 'Prio' => '300', 'Block' => "", 'Image' => 'search.png', 'NavBar' => "", 'Type' => "", 'AccessKey' => 's', 'Description' => 'Search', 'Name' => 'Search', 'Link' => 'Action=CustomerTicketSearch' }], 'NavBarName' => 'Ticket', 'Description' => 'Customer ticket search', 'Title' => 'Search' }; </pre>

B.2.38. Frontend::Customer::Preferences

B.2.38.1. CustomerPreferencesGroups###ClosedTickets

Description	Value
Description:	All parameters for the ClosedTickets object in the customer preferences.
Group:	Ticket
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerPreferencesGroups'}->{'ClosedTickets'} = { 'Prio' => '2000', 'Label' => 'Closed Tickets', 'DataSelected' => '1', 'PrefKey' => 'UserShowClosedTickets', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' => 'Show closed tickets.', 'Activ' => '1', 'Data' => { '1' => 'Yes', '0' => 'No' }, 'Colum' => 'Other Options' }; </pre>

B.2.38.2. CustomerPreferencesGroups###ShownTickets

Description	Value
Description:	All parameters for the ShownTickets object in the customer preferences.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerPreferencesGroups'}->{'ShownTickets'} = { 'Prio' => '4000', 'Label' => 'Shown Tickets', 'DataSelected' => '25', 'PrefKey' => 'UserShowTickets', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' => 'Max. shown Tickets a page in Overview.', 'Activ' => '1', 'Data' => { '25' => '25', '30' => '30', '20' => '20', '15' => '15' }, 'Colum' => 'Frontend' }; </pre>

B.2.38.3. CustomerPreferencesGroups###RefreshTime

Description	Value
Description:	All parameters for the RefreshTime object in the customer preferences.
Group:	Ticket
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerPreferencesGroups'}->{'RefreshTime'} = { 'Prio' => '4000', 'Label' => 'QueueView refresh time', 'PrefKey' => 'UserRefreshTime', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' => 'Select your QueueView refresh time.', 'Activ' => '1', 'Data' => { '' => 'off', '7' => '7 minutes', '10' => '10 minutes', '2' => '2 minutes', '15' => '15 minutes', '5' => '5 minutes' }, 'Colum' => 'Frontend' }; </pre>

B.3. FAQ

B.3.1. Core

B.3.1.1. FAQ::FAQHook

Description	Value
Description:	Default hook for FAQ.
Group:	FAQ
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'FAQ::FAQHook'} = 'FAQ';</code>

B.3.1.2. FAQ::Default::State

Description	Value
Description:	Default state for FAQ entry.
Group:	FAQ
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'FAQ::Default::State'} = 'internal (agent)';</code>

B.3.1.3. FAQ::Field1

Description	Value
Description:	Description of first FAQ field.
Group:	FAQ
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'FAQ::Field1'} = 'Symptom';</code>

B.3.1.4. FAQ::Field2

Description	Value
-------------	-------

Description	Value
Description:	Description of secound FAQ field.
Group:	FAQ
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'FAQ::Field2'} = 'Problem';</code>

B.3.1.5. FAQ::Field3

Description	Value
Description:	Description of third FAQ field.
Group:	FAQ
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'FAQ::Field3'} = 'Solution';</code>

B.3.1.6. FAQ::Field4

Description	Value
Description:	Description of fourth FAQ field.
Group:	FAQ
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'FAQ::Field4'} = 'field4';</code>

B.3.1.7. FAQ::Field5

Description	Value
Description:	Description of fiifth FAQ field
Group:	FAQ
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'FAQ::Field5'} = 'field5';</code>

B.3.1.8. FAQ::Field6

Description	Value
Description:	Description of sixth FAQ field.
Group:	FAQ
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'FAQ::Field6'} = 'Comment (internal)';</code>

B.3.2. Core::LinkObject**B.3.2.1. LinkObject###FAQ**

Description	Value
Description:	Link object settings for FAQ. Objects that are known by the system.
Group:	FAQ
SubGroup:	Core::LinkObject
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LinkObject'}->{'FAQ'} = { 'Type' => 'Object', 'LinkObjects' => ['Ticket', 'FAQ'], 'Name' => 'FAQ Object' };</code>

B.3.3. Frontend::Agent::ModuleRegistration**B.3.3.1. Frontend::Module###FAQ**

Description	Value
Description:	Frontend module registration for the FAQ object in the agent interface.
Group:	FAQ
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'FAQ'} = { 'NavBar' => [{ 'Prio' => '8300', 'Block' => 'ItemArea', 'GroupRo' => ['faq'], 'Image' => 'help.png', 'NavBar' => 'FAQ', 'Type' => 'Menu', 'AccessKey' => 'q', 'Description' => 'FAQ-Area', 'Name' => 'FAQ', 'Link' => 'Action=FAQ&Nav=Normal' }, { 'Prio' => '200', 'Block' => "", 'Image' => 'new.png', 'Group' => ['faq'], 'NavBar' => 'FAQ', 'Type' => "", 'AccessKey' => 'n', 'Description' => 'New Article', 'Name' => 'New Article', 'Link' => 'Action=FAQ&Subaction=Add' }, { 'Prio' => '300', 'Block' => "", 'GroupRo' => ['faq'], 'Image' => 'search.png', 'NavBar' => 'FAQ', 'Type' => "", 'AccessKey' => 's', 'Description' => 'FAQ-Search', 'Name' => 'Search', 'Link' => 'Action=FAQ&Subaction=Search' }, { 'Prio' => '310', 'Block' => "", 'GroupRo' => ['faq'], 'Image' => 'list.png', 'NavBar' => 'FAQ', 'Type' => "", 'AccessKey' => 'o', 'Description' => 'History', 'Name' => 'History', 'Link' => 'Action=FAQ&Subaction=SystemHistory' }], 'GroupRo' => ['faq'], 'NavBarName' => 'FAQ', 'Description' => 'FAQ-Area', 'Title' => "", 'Group' => ['faq'] }; </pre>

Description	Value
-------------	-------

B.3.3.2. Frontend::Module###FAQCategory

Description	Value
Description:	Frontend module registration for the FAQCategory object in the FAQ area.
Group:	FAQ
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'FAQCategory'} = { 'NavBar' => ['Prio' => '900', 'Image' => 'fileopen.png', 'NavBar' => 'FAQ', 'AccessKey' => 'g', 'Description' => 'Category', 'Name' => 'Category', 'Link' => 'Action=FAQCategory'], 'GroupRo' => [], 'NavBarName' => 'FAQ', 'Description' => 'FAQ-Category', 'Title' => 'Category', 'faq']; </pre>

B.3.3.3. Frontend::Module###FAQLanguage

Description	Value
Description:	Frontend module registration for the FAQLanguage object in the FAQ area.
Group:	FAQ
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'FAQLanguage'} = { 'NavBar' => [{ 'Prio' => '910', 'Block' => "", 'Image' => 'fileopen.png', 'NavBar' => 'FAQ', 'Type' => "", 'AccessKey' => 'u', 'Description' => 'Language', 'Name' => 'Language', 'Link' => 'Action=FAQLanguage' }], 'GroupRo' => [], 'NavBarName' => 'FAQ', 'Description' => 'FAQ-Language', 'Title' => 'Language', 'Group' => ['faq'] }; </pre>

B.3.4. Frontend::Customer::ModuleRegistration

B.3.4.1. CustomerFrontend::Module###CustomerFAQ

Description	Value
Description:	Frontend module registration for the CustomerFAQ object in the customer interface.
Group:	FAQ
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerFrontend::Module'}->{'CustomerFAQ'} = { 'NavBar' => [{ 'Prio' => '400', 'Block' => "", 'Image' => 'help.png', 'NavBar' => "", 'Type' => "", 'AccessKey' => 'f', 'Description' => 'FAQ-Area', 'Name' => 'FAQ-Area', 'Link' => 'Action=CustomerFAQ' }], 'NavBarName' => 'FAQ', 'Description' => 'Customer faq', 'Title' => "" }; </pre>

B.3.5. Frontend::Public

B.3.5.1. PublicFrontend::CommonParam###Action

Description	Value
Description:	Default value for the Action parameter for the public frontend. The Action parameter is used in the scripts of the system.
Group:	FAQ
SubGroup:	Frontend::Public
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PublicFrontend::CommonParam'}->{'Action'} = 'PublicFAQ';</code>

B.3.6. Frontend::Public::ModuleRegistration

B.3.6.1. PublicFrontend::Module###PublicFAQ

Description	Value
Description:	Frontend module registration for the PublicFAQ object in the public FAQ area.
Group:	FAQ
SubGroup:	Frontend::Public::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'PublicFrontend::Module'}->{'PublicFAQ'} = { 'NavBar' => ['Prio' => '400', 'Image' => 'help.png', 'NavBar' => "", 'AccessKey' => 'f', 'Description' => 'FAQ-Area', 'Name' => 'FAQ-Area', 'Link' => 'Action=CustomerFAQ'], 'NavBarName' => 'FAQ', 'Description' => 'Customer faq', };</code>

Appendix C. Credits

Like other open source projects we want to thank some people for their help and support. The following list is surely incomplete and we've definitely forgotten someone; sorry for that! Just drop us a note if you are not on this list.

The following persons have especially pushed the project or are still active supporters:

- Robert Kehl, who has created the Win32-installer for the 1.x releases. Thanks a lot, Robert!
- Torsten Werner, who maintains the Debian-installer for OTRS. Thanks a lot, Torsten, by making OTRS also available for the Debian community.
- Nils Jeppe (mirror Hamburg, Germany), Bryan Fullerton (mirror Toronto, Canada), Eberhard Mönkeberg (mirror Göttingen, Germany), Timo Dreger (mirror Düsseldorf, Germany) and Netmonic (mirror Vienna, Austria), who are mirroring our ftp server. Thanks a lot, with your help it is always possible to download OTRS quickly!
- Anja Schneider, who is helping with the translation of this manual and with the correction and revision of the German texts. Many thanks for your help and patience, Anja!
- We receive many ideas, Via the OTRS mailing lists, for system improvements, patches or bugfixes and great support for all users is available. Thanks a lot to all people on the mailing lists for your active assistance!

Also we want to give a big thank you to the following persons:

- 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
- Vladimir Gerdjikov
- Fred van Dijk
- Sebastien Guilbaud
- Wiktor Wodecki
- Arnold Ligtoet
- Antti Kämäräinen
- Nicolas Goralski
- Gilberto Cezar de Almeida
- Jorge Becerra
- Eddie Urenda
- Stella Power
- Andreas Haase
- Reiner Keller
- Covert Jake
- Moshe Leibovitch
- Björn Jacke
- Remo Catelotti
- Alfons Obermeyer
- Michael Rech
- Danie Theron
- Richard Paradies
- Art Powell

Appendix D. 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.