

# **OTRS 2.0 - Admin Manual**

## **OTRS 2.0 - Admin Manual**

Klong Jark Beach Edition

Copyright © 2005 OTRS GmbH

Richard Kammermeyer, Christian Schöpplein, Stefan Rother, Thomas Raith, Burchard Steinbild, Andre Mindermann, 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

<b>Preface .....</b>	<b>i</b>
<b>1. Basics about trouble ticket systems.....</b>	<b>1</b>
1.1. What is a trouble ticket system, when do you need it? .....	1
1.2. What is a trouble ticket?.....	2
<b>2. OTRS - Open Ticket Request System .....</b>	<b>3</b>
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
<b>3. Installation of the OTRS framework .....</b>	<b>8</b>
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 .....	19
3.2. Installation from source (Linux, Unix) .....	20
3.2.1. Preparing the installation from source.....	20
3.2.2. Installation of needed perl modules.....	21
3.2.3. Configuring the apache web server .....	23
3.2.4. Configuring the database .....	24
3.2.5. Setting up the cron jobs for OTRS .....	29
<b>4. First steps in OTRS.....</b>	<b>34</b>
4.1. Agent web interface .....	34
4.2. Customer web interface.....	35
4.3. Public web interface.....	35
4.4. First login .....	36
4.5. An overview to the web interface.....	37
4.6. What is a queue? .....	40
4.7. User preferences.....	41
<b>5. The admin area of OTRS .....</b>	<b>44</b>
5.1. Basics .....	44
5.2. User, groups and roles .....	45
5.2.1. User.....	45
5.2.2. Groups .....	46
5.2.3. Roles .....	48
5.3. Customer users and customer groups.....	51
5.3.1. Customer users .....	51
5.3.2. Customer groups.....	52
5.4. Queues.....	53
5.5. Salutations, signatures, attachments and responses .....	55
5.5.1. Salutations .....	57
5.5.2. Signatures .....	58
5.5.3. Attachments.....	59
5.6. Auto answers.....	61

5.7. Email addresses .....	63
5.8. Notifications .....	64
5.9. SMIME.....	65
5.10. PGP .....	66
5.11. Statess.....	67
5.12. The graphical configuration frontend (SysConfig) .....	68
5.13. Using POP3 accounts.....	69
5.14. Filtering incoming messages.....	70
5.15. Executing automated jobs with the GenericAgent.....	73
5.16. Admin email.....	75
5.17. Session management .....	76
5.18. System Log .....	77
5.19. SQL queries via the select box.....	78
5.20. Package manager.....	79
<b>6. Configuring the system.....</b>	<b>81</b>
6.1. The config files of OTRS .....	81
6.2. Configuring the system through the web interface .....	81
<b>7. Sending/Receiving emails.....</b>	<b>84</b>
7.1. Sending emails .....	84
7.1.1. Via Sendmail (default).....	84
7.1.2. Via SMTP relay/smarthost .....	84
7.2. Receiving emails .....	84
7.2.1. Via POP3 accounts - the easy way (PostMasterPOP3.pl) .....	85
7.2.2. Via command line program and e. g. procmail (PostMaster.pl).....	86
7.2.3. Fetching emails via POP3 or IMAP and fetchmail for PostMaster.pl.....	87
7.2.4. Filtering/dispatching by OTRS/PostMaster modules (for more complex dispatching)87	
<b>8. Using external backends.....</b>	<b>90</b>
8.1. Customer data .....	90
8.2. Customer user backend .....	90
8.2.1. Database (Default).....	90
8.2.2. LDAP.....	93
8.3. Customer authentication backend .....	95
8.3.1. Database (Default).....	95
8.3.2. LDAP .....	95
8.3.3. HTTPBasicAuth .....	96
8.3.4. Radius .....	96
8.4. Customize the customer self registration .....	96
8.4.1. Customize the web interface .....	97
8.4.2. Customer mapping.....	97
8.4.3. Altering the customer table .....	98
<b>9. Additional applications .....</b>	<b>99</b>
9.1. Calendar .....	99
9.2. Content manager .....	99
9.3. File manager.....	99
9.4. Web mailer .....	100
9.5. FAQ .....	101

9.6. System status .....	101
<b>10. Performance Tuning .....</b>	<b>103</b>
10.1. OTRS .....	103
10.1.1. TicketIndexModule .....	103
10.1.2. TicketStorageModule .....	103
10.2. Database .....	104
10.2.1. MySQL .....	104
10.2.2. PostgreSQL .....	105
10.3. Webserver .....	105
10.3.1. Preestablish datababase connections .....	105
10.3.2. Preloaded modules - startup.pl .....	105
10.3.3. Reload perl modules when updated on disk .....	105
10.3.4. Choosing the Right Strategy .....	106
10.3.5. mod_gzip/mod_deflate .....	106
10.3.6. mod_dosevasive .....	106
<b>11. Backing up the system .....</b>	<b>107</b>
11.1. Backup .....	107
11.2. Restore .....	108
<b>A. Additional Resources .....</b>	<b>109</b>
A.1. Homepage OTRS.org .....	109
A.2. Mailinglists .....	109
A.3. Bugtracking .....	110
A.4. Commercial Support .....	111
<b>B. Config Referenzliste .....</b>	<b>112</b>
B.1. Framework .....	112
B.1.1. Core .....	112
B.1.2. Core::Log .....	119
B.1.3. Core::MIME-Viewer .....	121
B.1.4. Core::Package .....	122
B.1.5. Core::Sendmail .....	124
B.1.6. Core::Session .....	126
B.1.7. Core::SpellChecker .....	130
B.1.8. Core::Time .....	132
B.1.9. Core::Web .....	135
B.1.10. Crypt::PGP .....	137
B.1.11. Crypt::SMIME .....	139
B.1.12. Frontend::Admin::ModuleRegistration .....	140
B.1.13. Frontend::Agent .....	150
B.1.14. Frontend::Agent::Auth::LDAP .....	153
B.1.15. Frontend::Agent::ModuleNotify .....	154
B.1.16. Frontend::Agent::ModuleRegistration .....	155
B.1.17. Frontend::Agent::Preferences .....	159
B.1.18. Frontend::Customer .....	163
B.1.19. Frontend::Customer::Auth .....	168
B.1.20. Frontend::Customer::ModuleNotify .....	174
B.1.21. Frontend::Customer::ModuleRegistration .....	175

B.1.22. Frontend::Customer::Preferences .....	177
B.2. Ticket.....	181
B.2.1. Core::LinkObject .....	181
B.2.2. Core::PostMaster .....	181
B.2.3. Core::Stats.....	189
B.2.4. Core::Ticket .....	191
B.2.5. Core::TicketFreeText .....	201
B.2.6. Frontend::Admin::ModuleRegistration .....	210
B.2.7. Frontend::Agent.....	217
B.2.8. Frontend::Agent::ModuleNotify .....	226
B.2.9. Frontend::Agent::ModuleRegistration.....	226
B.2.10. Frontend::Agent::NavBarModule .....	238
B.2.11. Frontend::Agent::Preferences .....	238
B.2.12. Frontend::Agent::Ticket::ArticleAttachmentModule .....	243
B.2.13. Frontend::Agent::Ticket::ArticleComposeModule.....	244
B.2.14. Frontend::Agent::Ticket::ArticleViewModule.....	244
B.2.15. Frontend::Agent::Ticket::ArticleViewModulePre .....	245
B.2.16. Frontend::Agent::Ticket::MenuModule.....	246
B.2.17. Frontend::Agent::Ticket::MenuModulePre .....	251
B.2.18. Frontend::Agent::Ticket::ViewBounce .....	254
B.2.19. Frontend::Agent::Ticket::ViewClose .....	255
B.2.20. Frontend::Agent::Ticket::ViewCompose .....	257
B.2.21. Frontend::Agent::Ticket::ViewEmailNew .....	258
B.2.22. Frontend::Agent::Ticket::ViewForward.....	261
B.2.23. Frontend::Agent::Ticket::ViewHistory .....	262
B.2.24. Frontend::Agent::Ticket::ViewMerge.....	262
B.2.25. Frontend::Agent::Ticket::ViewMove .....	263
B.2.26. Frontend::Agent::Ticket::ViewNote .....	264
B.2.27. Frontend::Agent::Ticket::ViewOwner .....	266
B.2.28. Frontend::Agent::Ticket::ViewPending .....	267
B.2.29. Frontend::Agent::Ticket::ViewPhone .....	268
B.2.30. Frontend::Agent::Ticket::ViewPhoneNew .....	271
B.2.31. Frontend::Agent::Ticket::ViewQueue.....	274
B.2.32. Frontend::Agent::Ticket::ViewSearch .....	277
B.2.33. Frontend::Agent::Ticket::ViewStatus .....	280
B.2.34. Frontend::Agent::Ticket::ViewZoom.....	280
B.2.35. Frontend::Customer .....	282
B.2.36. Frontend::Customer::ModuleRegistration .....	290
B.2.37. Frontend::Customer::Preferences .....	294
B.3. FAQ.....	295
B.3.1. Core.....	296
B.3.2. Core::LinkObject .....	298
B.3.3. Frontend::Agent::ModuleRegistration.....	298
B.3.4. Frontend::Customer::ModuleRegistration .....	301
B.3.5. Frontend::Public.....	301
B.3.6. Frontend::Public::ModuleRegistration .....	302

<b>C. Credits.....</b>	<b>303</b>
<b>D. GNU Free Documentation License.....</b>	<b>305</b>
0. PREAMBLE .....	305
1. APPLICABILITY AND DEFINITIONS .....	305
2. VERBATIM COPYING.....	306
3. COPYING IN QUANTITY .....	306
4. MODIFICATIONS.....	307
5. COMBINING DOCUMENTS.....	308
6. COLLECTIONS OF DOCUMENTS .....	309
7. AGGREGATION WITH INDEPENDENT WORKS.....	309
8. TRANSLATION .....	310
9. TERMINATION.....	310
10. FUTURE REVISIONS OF THIS LICENSE.....	310
How to use this License for your documents .....	310

## List of Tables

3-1. Needed perl modules for OTRS .....	21
3-2. Description of the several cron job scripts .....	30
5-1. Default groups after OTRS has been installed .....	46
5-2. Rights in the user groups of OTRS.....	48
5-3. Events for auto answers.....	62
5-4. Function of the different X-OTRS-headers .....	71
A-1. Mailinglists .....	109

## List of Examples

5-1. Aussortierung von Spammails in eine bestimmte Queue.....	72
7-1. .fetchmailrc.....	87
7-2. Example jobs for the filter module Kernel::System::PostMaster::Filter::Match.....	88
7-3. Example job for the filter module Kernel::System::PostMaster::Filter::CMD .....	88



# Preface

This book wants to make live 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 liters of coffee and several pizzas were invested into the following sections, it is still not complete. For sure, there will be errors, parts with complicated explanations or still missing chapters about important things. You can be sure that 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 us if you miss something 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 fabricant who produces video recorders. Since programing video recorders is very complex, Max gets many mails from customers who need help with the devices. At some days, it may happen that Max has not enough time to answer all mails so that his customers has to be patient till 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 it may happen that the developers Joe and John have to help Max to answer the mails because Max can't reply fast enough to all 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 one desperate customer. So it happens that Joe answers the first and John the second mail of a customer, both with another helpful hint 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 replys of Joe and John nor does he know all problems the customers have had with the video recorders nor which problems occure very often nor how much time and money he has to spend for the 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 not longer send 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 for every single request, the ticket number. Now, the customers are happy because they quickly got a response to their requests and it is not necessary to send a secound 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.

Lets imagine that Mr Schmith makes a request to Max's company and his message is processed by

OTRS. John answers this request in a short way but Mr Smith wants to know more so that he replies again to John's mail. Since John has other things to do, Max answers to 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.

Ofcourse, 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 what 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 if the patient visits the hospital for the first time. All important data about the patient, as personal data and facts about the medical problem, are noticed in the medical report. Each doctor adds information about the illness and the medication to the medical report 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 emails. The messages are saved in the system. When a customer sends a request, a new ticket is generated by the system like a new medical report is opened. The answer to this new ticket can be compared to a doctor's entry into the medical report, a new answer of the customer 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 specific informations. 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, date, 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 can also find information about the hard- and software requirements of the ticket system. If you need commercial support for OTRS or if you want to contact the community, this chapter tells you how to do this.

## 2.1. Basics

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

OTRS is separated in 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 which are included in the central framework.

### **The features of OTRS**

- Webinterface:
  - Easy and initial handling with a web browser.
  - Because no active web contents like Flash or Java applets are used, the web interface is useable with nearly every web browser, 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 own tickets.
  - The web interface can be customized by different themes, own themes can be integrated.
  - Support for many languages.
  - The look of output templates can be customized (dtl).
  - Mails from and into the system can contain several attachments.

- Mail interface:
  - Support for mail attachments (MIME support).
  - Automatic conversion of HTML into plain text messages (more security for dangerous content and faster searchable).
  - Mails can be filtered with the X-OTRS headers of the system or via mail addresses, e.g. to sort out spam messages.
  - PGP support, creation and import of own keys, signing and encrypting outgoing mails, 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 "Generic-Agent".
  - 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 oder OpenLDAP.
- Setting up an own ticket identifier, e.g. Cal#, Ticket#, Request#, e.g.
- The integration of an 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.
- The authentication for customers can be realized 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 about up-dates via mail or the web interface.
- Escalation for tickets.
- Support for different time zones.
- Simple integration of own addons 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 oder FreeBSD) but on all Microsoft Windows platforms, too. OTRS has no big hardware requirements. We recommend to use a machine with at least a 2 GHz Pentium IV or comparable CPU, 512 MB RAM and a 20 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 like OTRS. The database back-end can be installed on the local or on an other host.

For the web server, we recommend to use apache 1.3.x or apache 2.x. With this web server, you can use the mod\_perl module which improves the performance of OTRS rapidly. If you can't use apache, OTRS should run on every other web server that is able to 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 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.

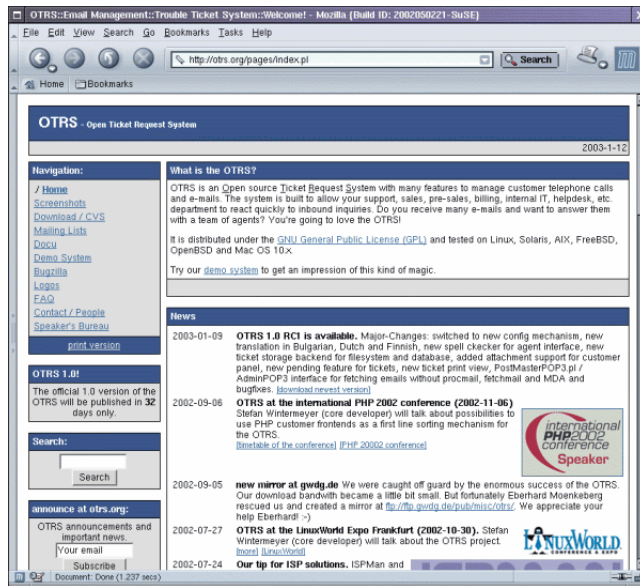
For perl, we recommend to use at least version 5.8. You need some additional modules which can be installed either with the perl shell and CPAN or via the package manager of your operating system (rpm, yast, apt-get).

The section about the manual installation of perl modules needed describes in more details 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 to the perl modules needed.

## 2.4. Community

OTRS has a big user community. Via mailinglists users and developers discuss about OTRS and exchange information on the trouble ticket system. 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 at otrs.com](mailto:sales@otrs.com) ([mailto:sales at 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:~ #
```

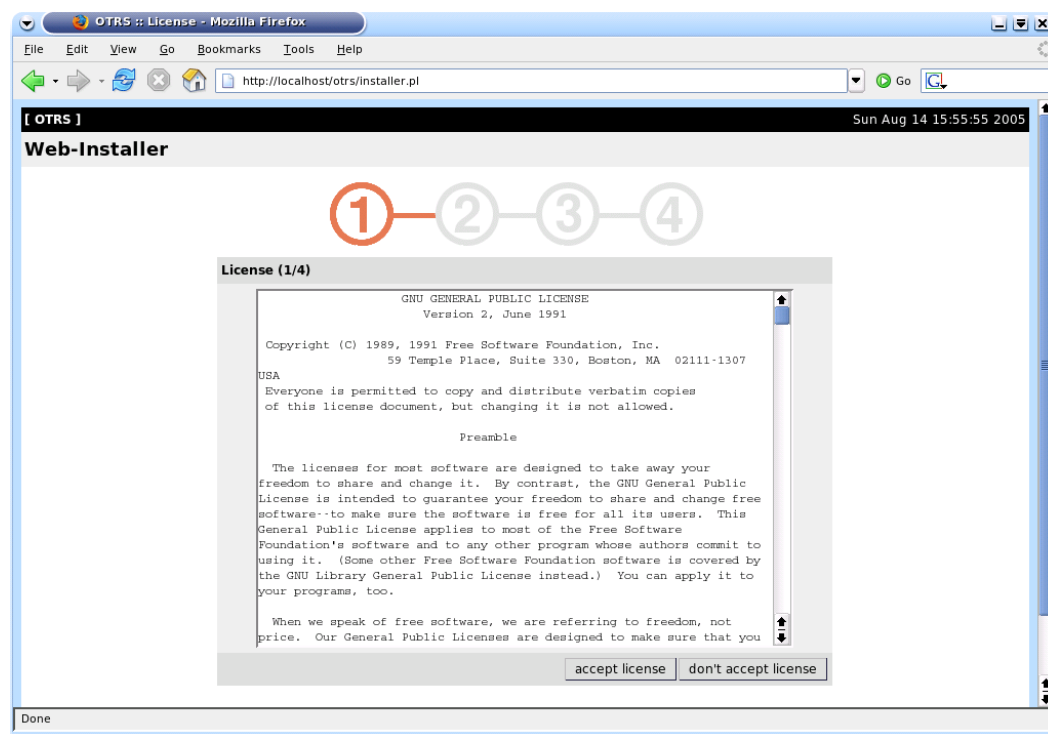
The installation of the OTRS rpm is finished. Restart your webserver to load the OTRS specific changes in your web server configuration:

```
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 title bar reads "OTRS :: Create Database - Mozilla Firefox". The address bar shows "http://localhost/otrs/installer.pl". The page header includes "[ OTRS ]" and a timestamp "Sun Aug 14 15:56:54 2005". The main heading is "Web-Installer". A progress indicator at the top shows four steps: 1 (green), 2 (red and highlighted), 3 (grey), and 4 (grey). The current step is "Create Database (2/4)". The form contains the following fields and options:

- Admin-User:** root
- Admin-Password:** (empty field)
- Host:** localhost
- Type:** MySQL (dropdown menu)
- Database-User (New):**
  - User:** otrs
  - Password:** \*\*\*
  - DB connect host:** localhost
- Database:**
  - Name:** otrs
  - Action:** Create (selected) - Delete

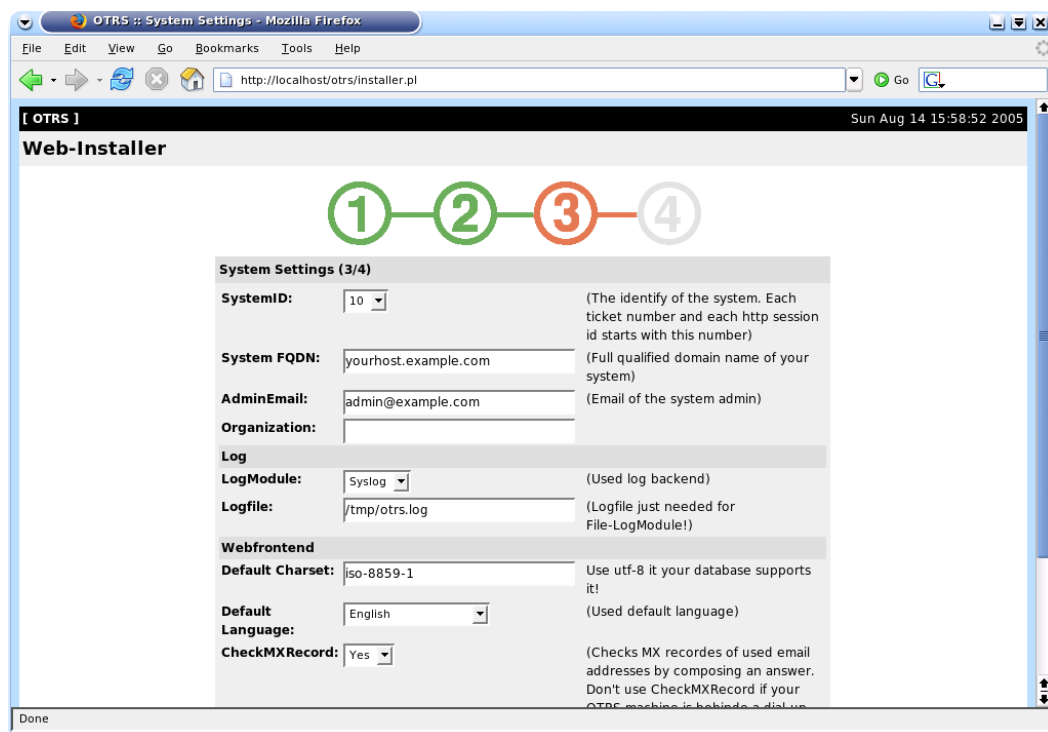
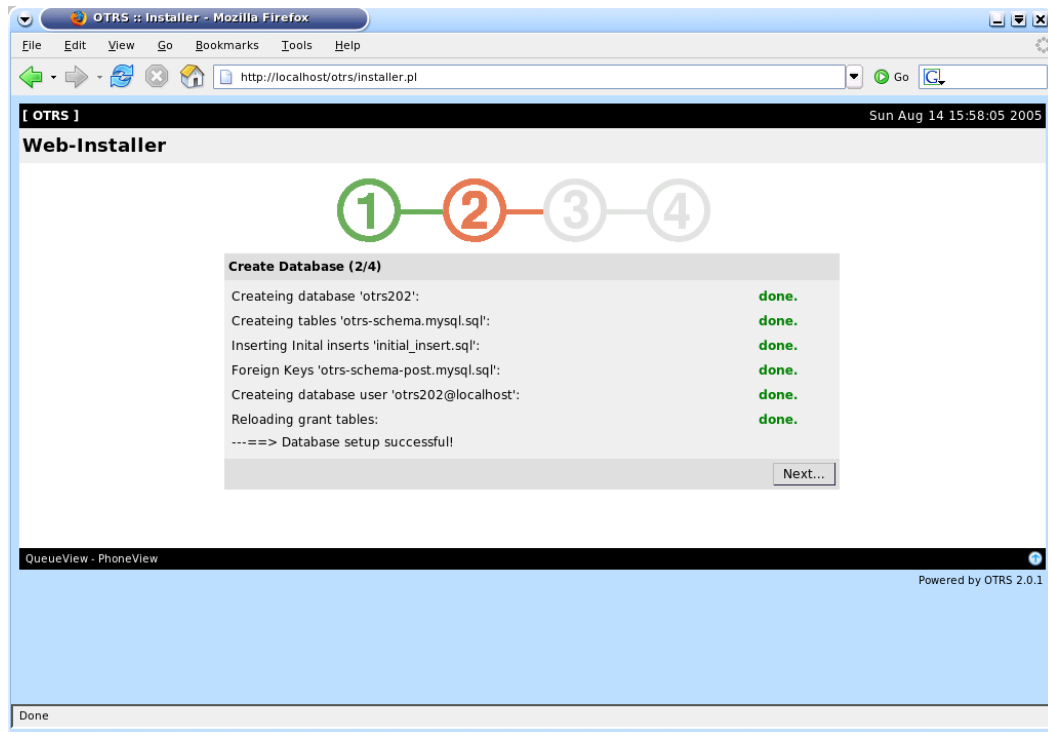
Red text annotations provide additional information:

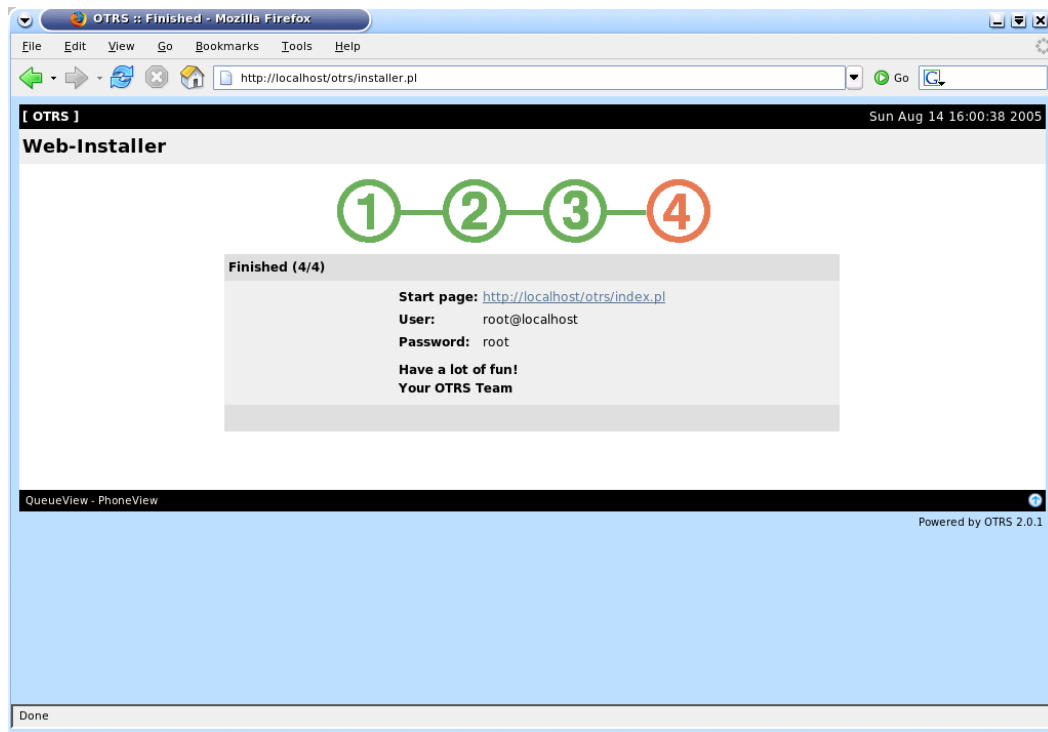
- Next to the Admin-Password field: "(\* your MySQL DB should have a root password! Default is empty!)"
- Next to the Database-User Password field: "(default 'hot')"

A "Next..." button is located 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

This section Describes the installation of OTRS on Debian 3.1 (Sarge).

Install OTRS with apt-get via the command line. OTRS needs some other packages to be installed, for example a database server, a web server or some perl modules. Let apt-get take care of these dependences. Execute the following command to install OTRS:

```
linux:~# apt-get install otrs
Reading Package Lists... Done
Building Dependency Tree... Done
Suggested packages:
  otrs-doc
The following NEW packages will be installed:
  otrs
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 947kB of archives.
After unpacking 6132kB of additional disk space will be used.
Get:1 ftp://ftp.uni-erlangen.de sarge/main otrs 1.3.2p01-5 [947kB]
Fetched 947kB in 10s (93.7kB/s)
Selecting previously deselected package otrs.
(Reading database ... 73351 files and directories currently installed.)
Unpacking otrs (from ../otrs_1.3.2p01-5_all.deb) ...
Setting up otrs (1.3.2p01-5) ...
Adding system user 'otrs'...
Adding new user 'otrs' (105) with group 'www-data'.
```

```
Not creating home directory.
SetPermissions.sh <$Revision: 1.4 $> - set OTRS file permissions
Copyright (c) 2001-2003 Martin Edenhofer <martin@otrs.org>
Setting file permissions...
chown -R 0:0 /usr/share/otrs
chown otrs:www-data /usr/share/otrs
chown -R otrs:www-data /usr/share/otrs/var/
chown -R www-data:www-data /usr/share/otrs/var/sessions/
touch && chown otrs:www-data
/usr/share/otrs/var/log/TicketCounter.log
chmod -R 755 /usr/share/otrs/bin/
(chown && chmod 700) otrs:0
/usr/share/otrs/bin/DeleteSessionIDs.pl
(chown && chmod 700) otrs:0 /usr/share/otrs/bin/UnlockTickets.pl
(chown && chmod 700) otrs:0 /usr/share/otrs/bin/otrs.getConfig

linux:~#
```

After the installation of the OTRS package you need to configure the webserver for the trouble ticket system (on debian systems this is not done automatically).

**Important:** The following description is based on the fact, that you use the web server apache 2.x. If you are using apache 1.3.x you have to do the same things. The configuration files of apache 1.3.x are located at `/etc/apache`.

Create the file `otrs.conf` and insert the following lines. Save the file into the directory `/etc/apache2/conf.d`.

```
#
# Basic configuration file for otrs on debian
#
# agent, admin and customer frontend
#
ScriptAlias /otrs/ "/usr/share/otrs/bin/cgi-bin/"
Alias /otrs-web/ "/usr/share/otrs/var/httpd/htdocs/"
#
```

Restart your webserver to use the new config settings.

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

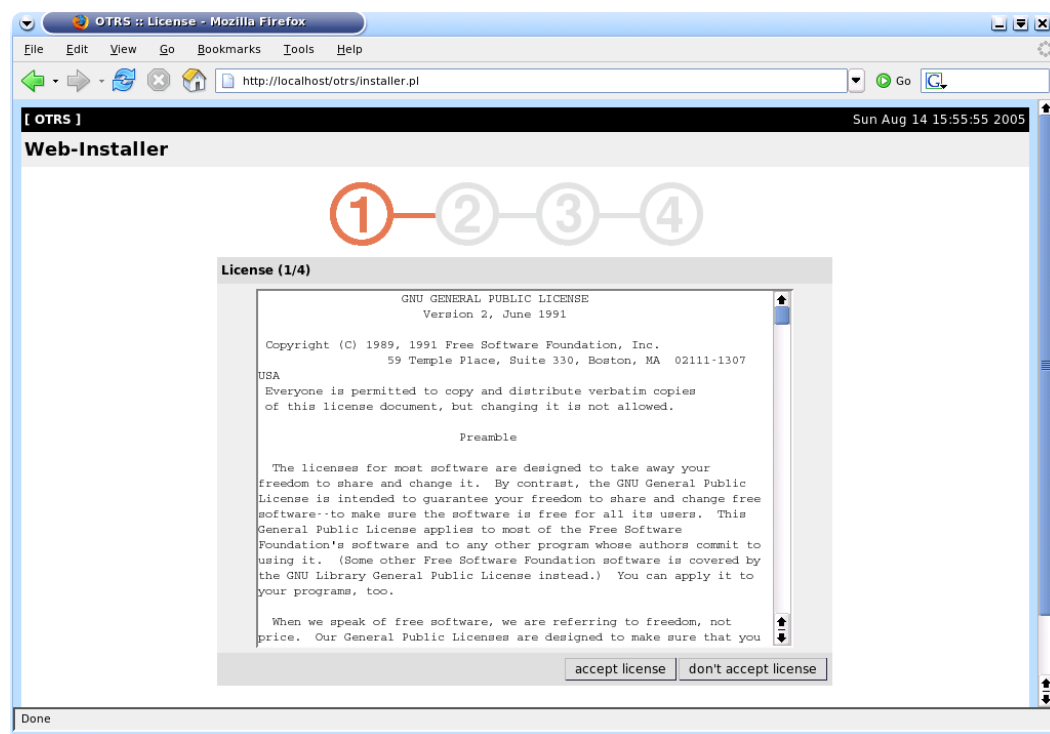


```
linux:~#
```

In the next step you have to setup the OTRS database. If you use MySQL for 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 date/time 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

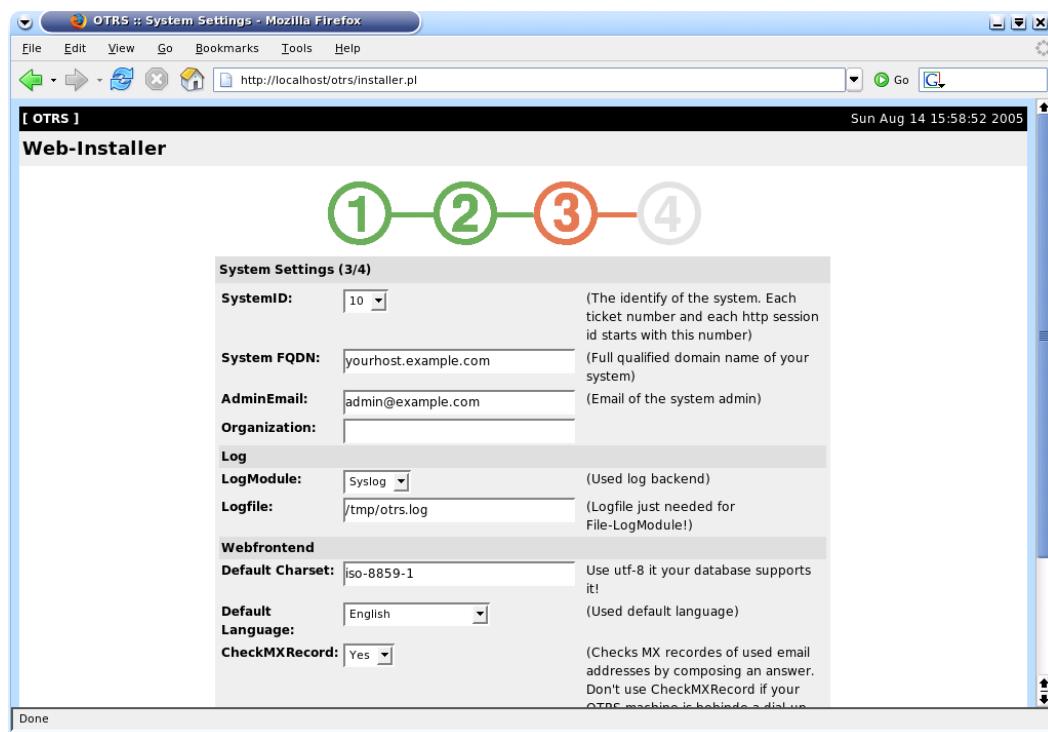
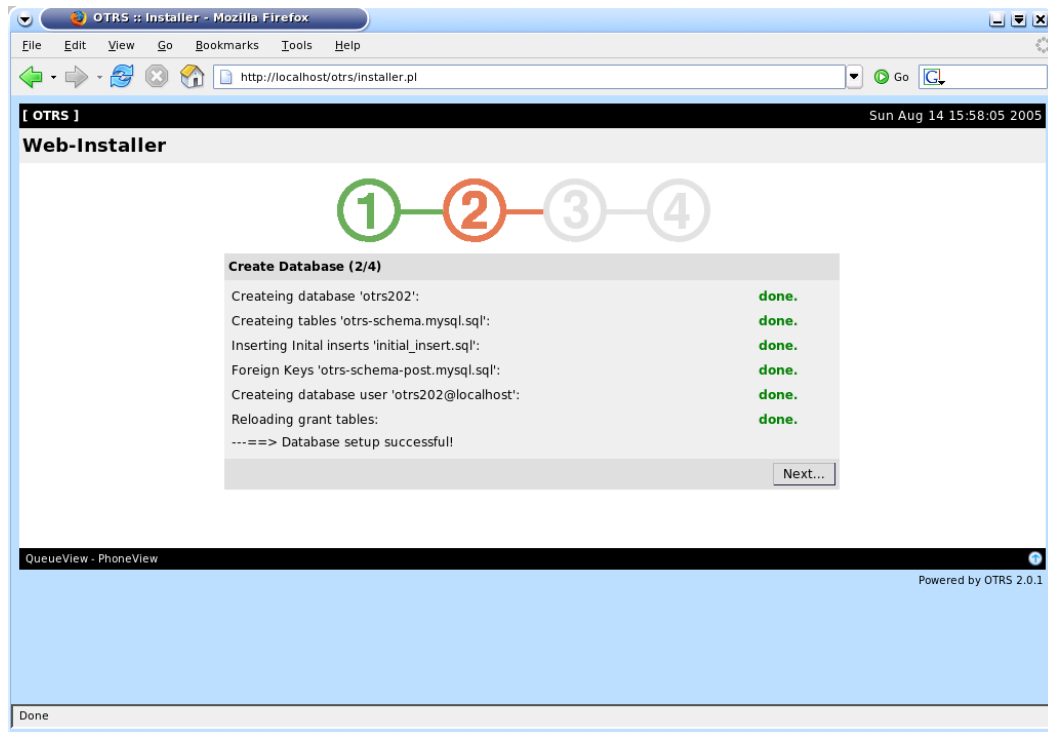
Red text annotations on the right side of the form:

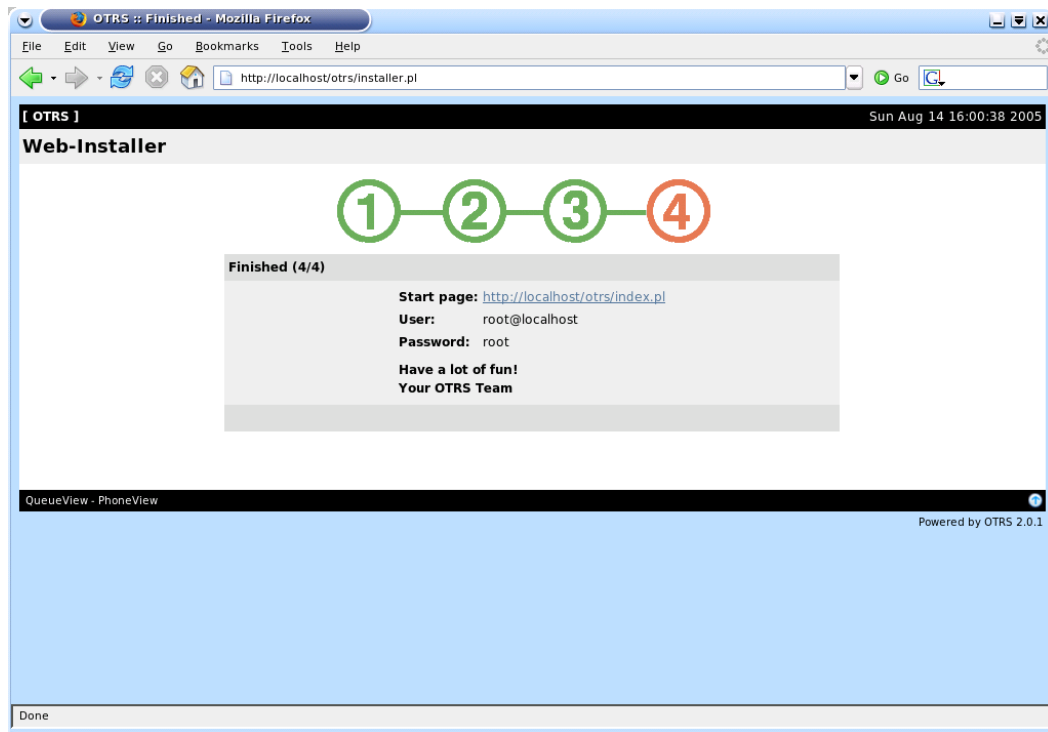
- Next to the Admin-Password field: "(\* your MySQL DB should have a root password! Default is empty!)"
- Next to the Database-User Password field: "(default 'hot')"

A "Next..." button is located at the bottom right of the form. At the bottom of the browser window, a status bar 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!





Now the installation of OTRS is finished. You should be able to use the web interface of OTRS. The cron jobs, which are needed for OTRS, are automatically installed and configured. Log in as OTRS admin and configure the system for your needs. Go to the address <http://localhost/otrs/index.pl> (<http://localhost/otrs/index.pl>) with a web browser and use "root@localhost" as username and "root" as password to access the web interface of the ticket system.

### Warning

Please change this password as soon as possible too! It is also a default password!

### 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.

Name	Description
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.
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 /opt/otrs/bin/cgi-bin/index.pl** and **perl -cw /opt/otrs/bin/PostMaster.pl**. If the output of both commands is "syntax OK", you have a proper perl installation to use with OTRS.

```

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

```

### 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 configuration file for otrs on debian
#
# agent, admin and customer frontend
#
ScriptAlias /otrs/ "/opt/otrs/bin/cgi-bin/"
Alias /otrs-web/ "/opt/otrs/var/httpd/htdocs/"
#
```

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`**.

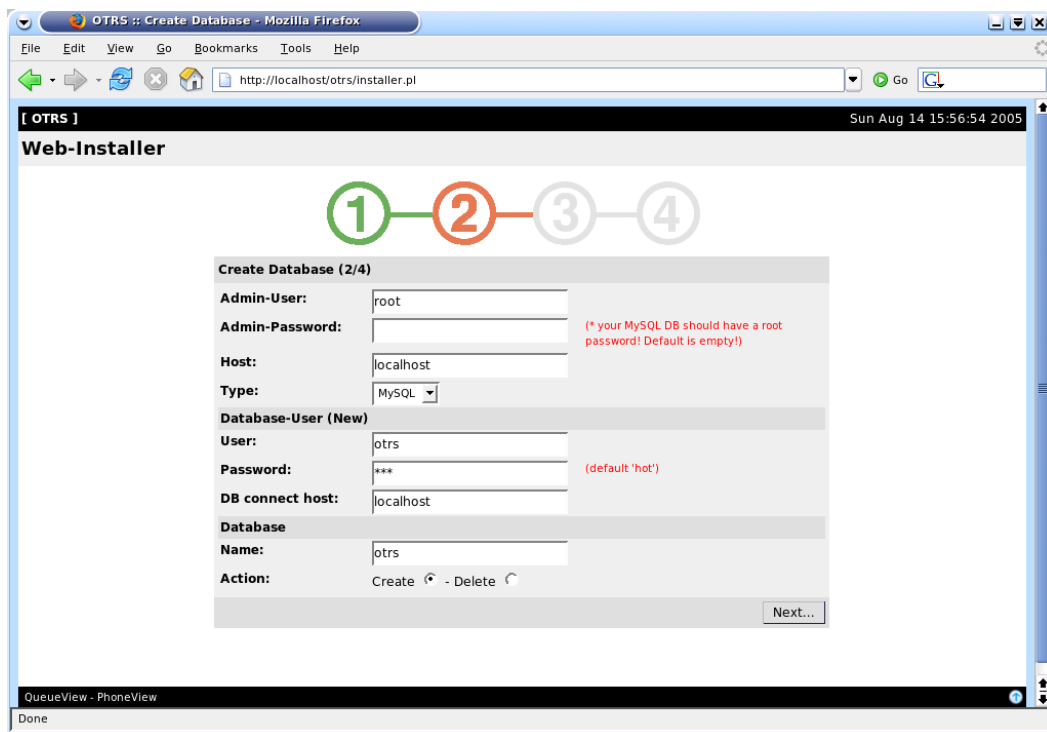
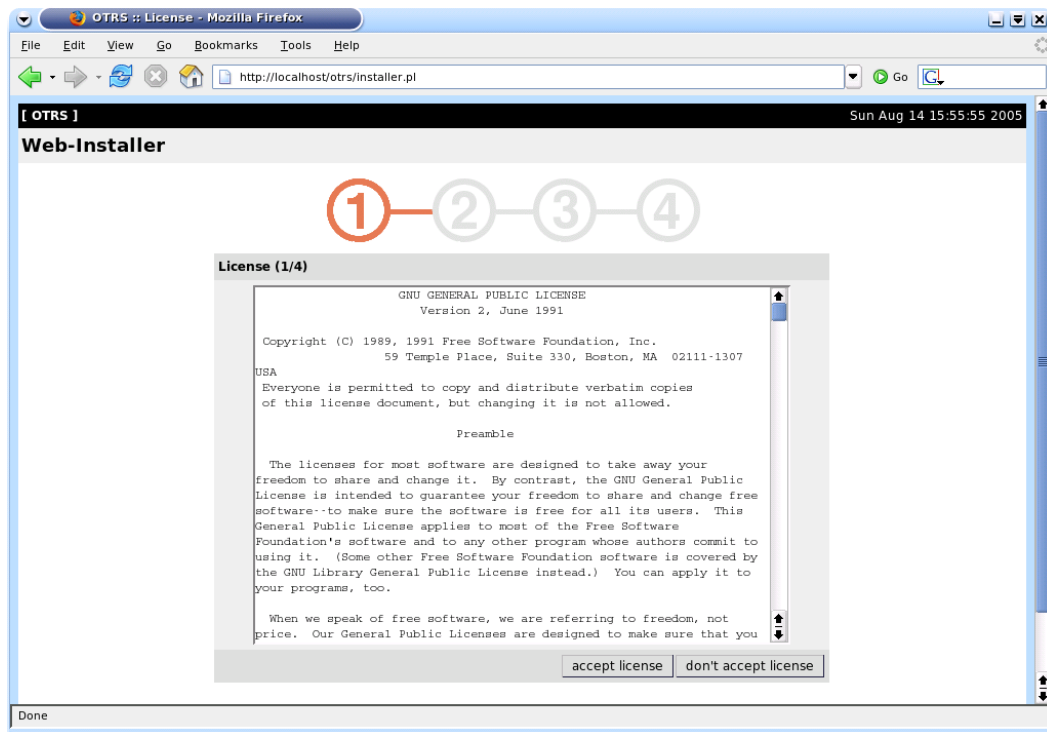
```
linux:/etc/apache2/conf.d# /etc/init.d/apache2 restart
Forcing reload of web server: Apache2.
linux:/etc/apache2/conf.d#
```

Now your webserver should be basicaly configured for OTRS.

## 3.2.4. Configuring the database

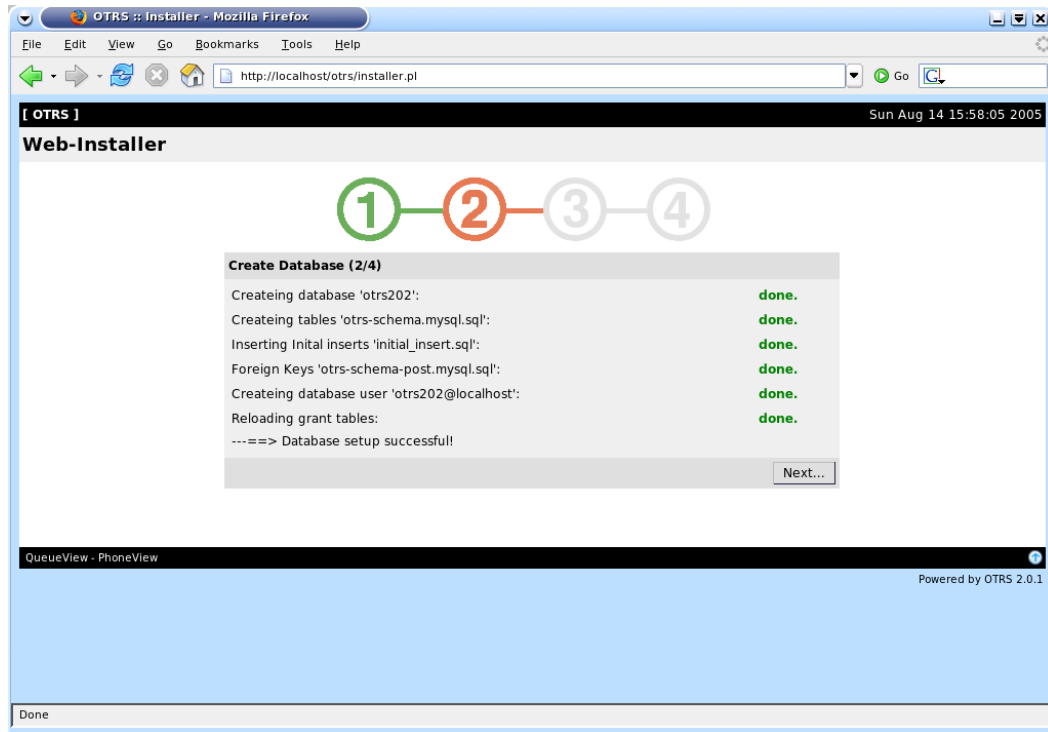
### 3.2.4.1. The simple way - Using the web installer (works only with MySQL)

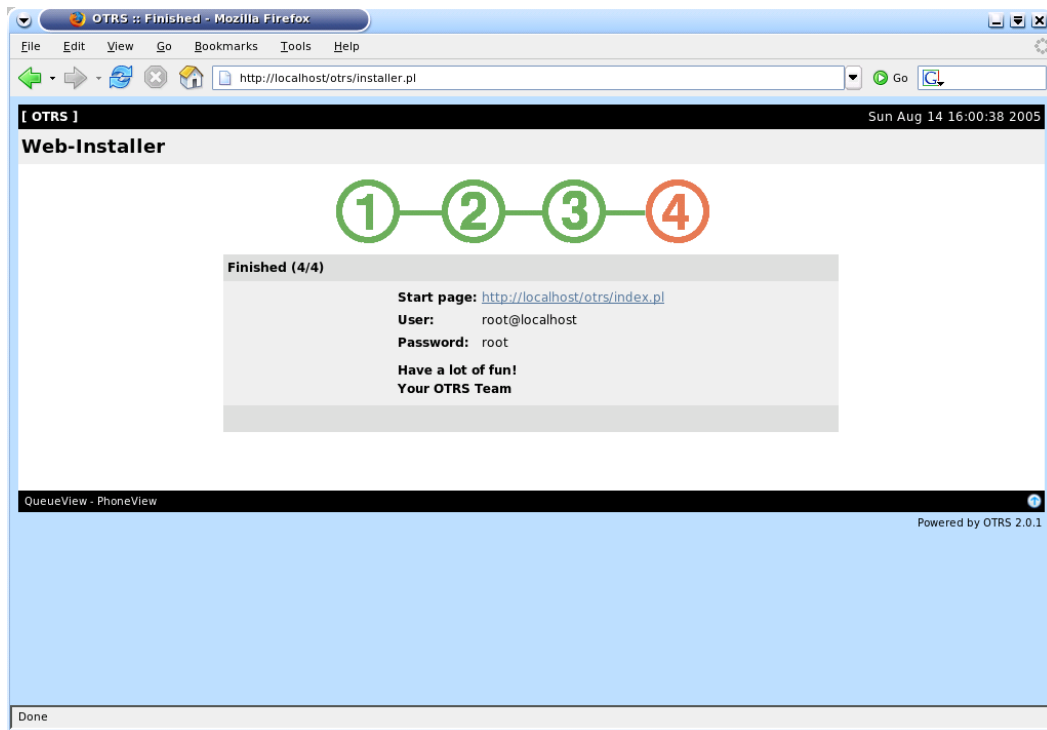
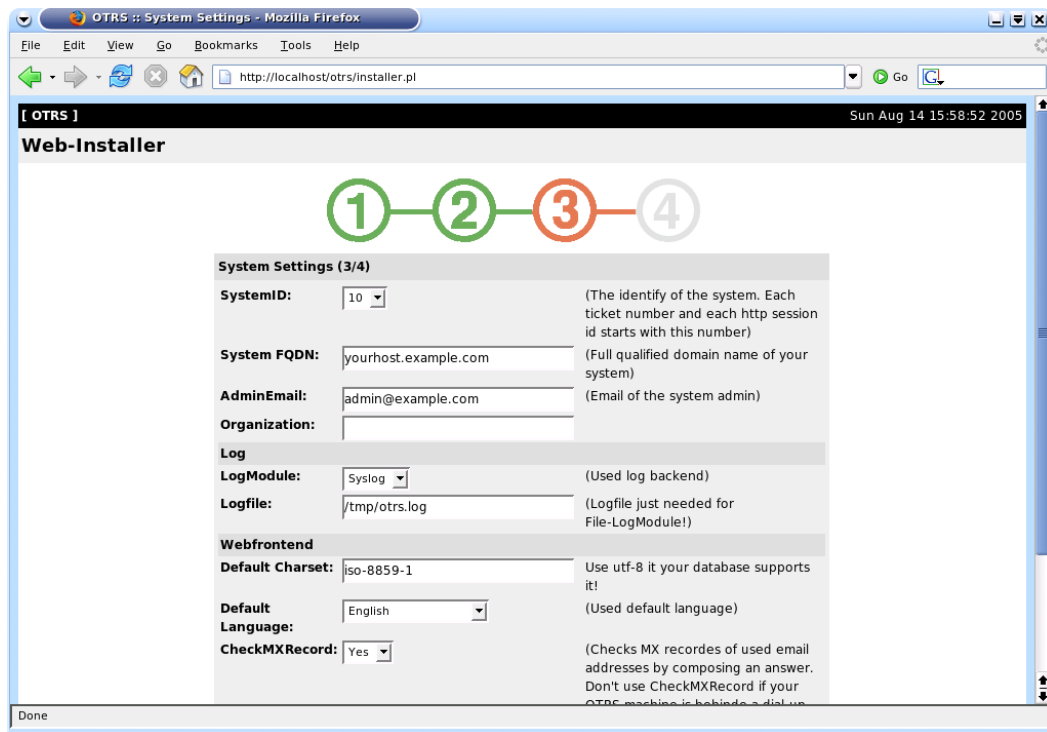
If you are using MySQL for database back-end, you can easily configure the OTRS database via a web front-end. Use the URL `http://localhost/otrs/installer.pl` (`http://localhost/otrs/installer.pl`) to access the start page of the web installer. Just follow the few steps through the installation process.



## 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:

```
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: 1.4 $>
Copyright (c) 2002 Martin Edenhofer <martin@otrs.org>
(using /opt/otrs) done
linux:~/bin$ 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 1.4 2005/10/06 13:50:16 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 1.4 2005/10/06 13:50:16 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 1.4 2005/10/06 13:50:16 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 1.4 2005/10/06 13:50:16 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 1.4 2005/10/06 13:50:16 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 1.4 2005/10/06 13:50:16 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 1.4 2005/10/06 13:50:16 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 1.4 2005/10/06 13:50:16 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 1.4 2005/10/06 13:50:16 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 1.4 2005/10/06 13:50:16 cs Exp $
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# unlock every hour old locked tickets
35 * * * * $HOME/bin/UnlockTickets.pl --timeout >> /dev/null

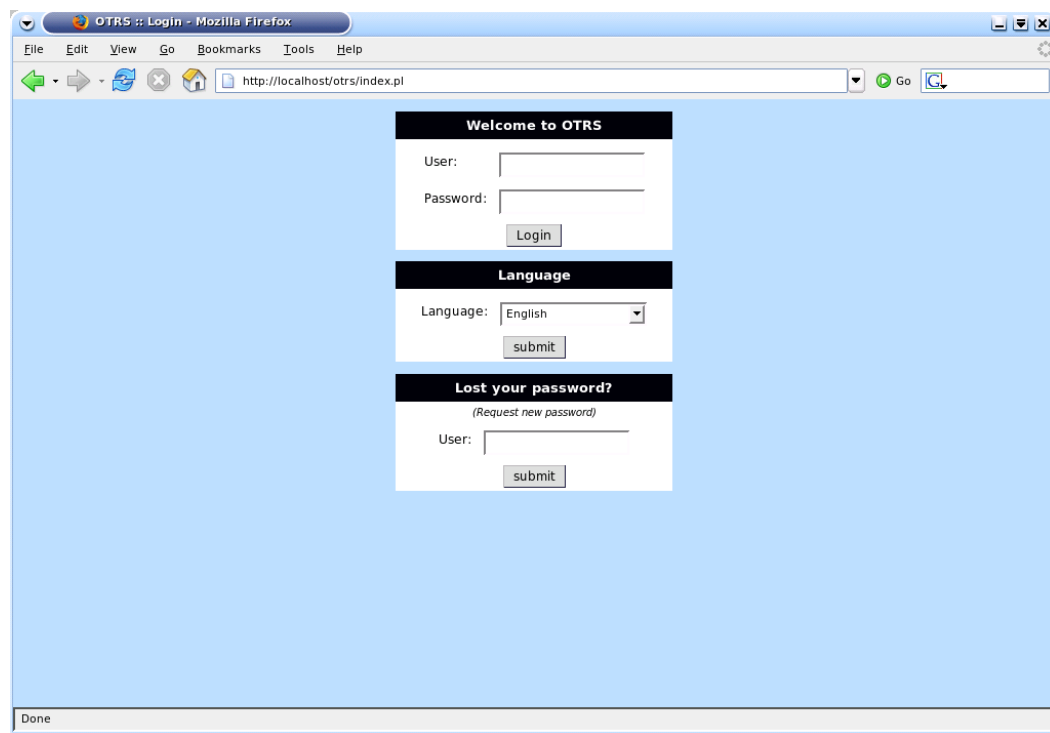
linux:/opt/otrs/bin#
```

# 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.

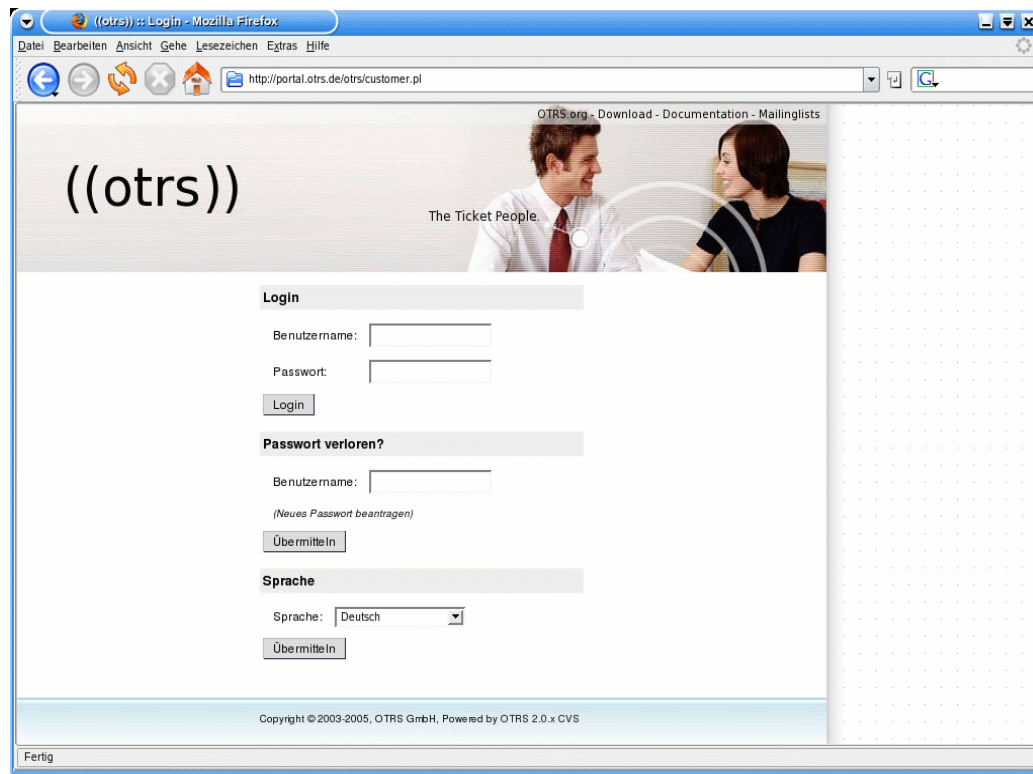


Lets say that your OTRS host is reachable via the URL *http://www.example.com* (*http://www.example.com/*) then the agents and also the OTRS administrator can reach the login screen

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.

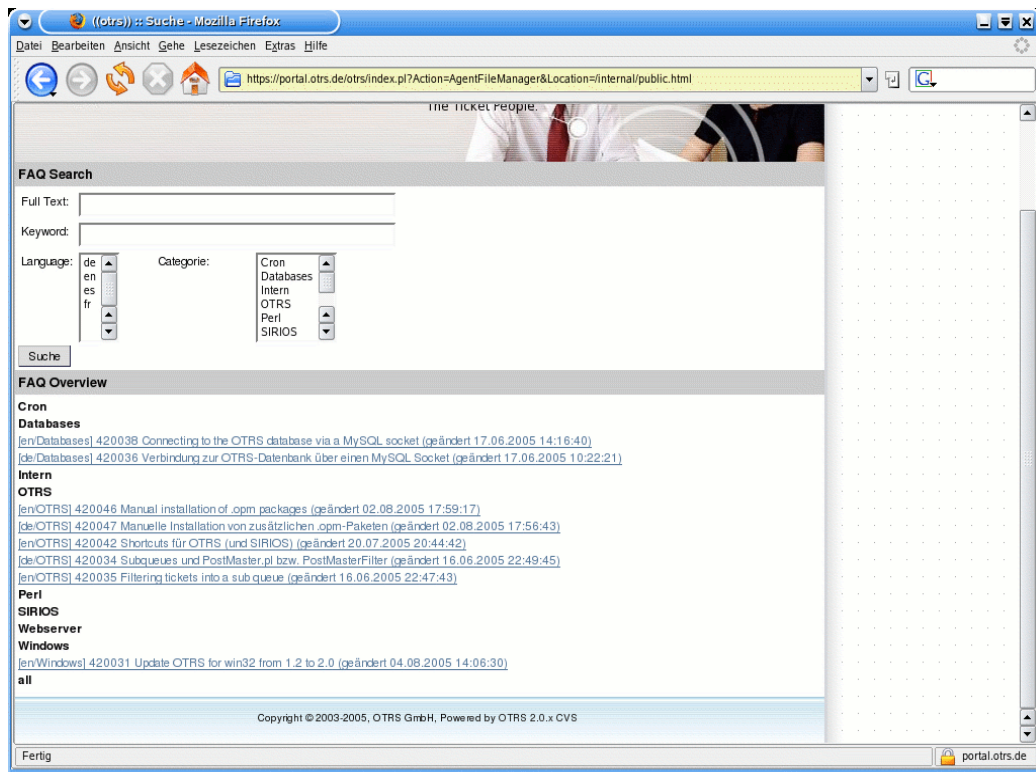


Lets say the host with the OTRS installation is reachable via the domain <http://www.example.com> (<http://www.example.com/>) then the login screen for customers can be reached by the URL <http://www.example.com/otrs/customer.pl> (<http://www.example.com/otrs/customer.pl>) with a web browser.

## 4.3. Public web interface

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

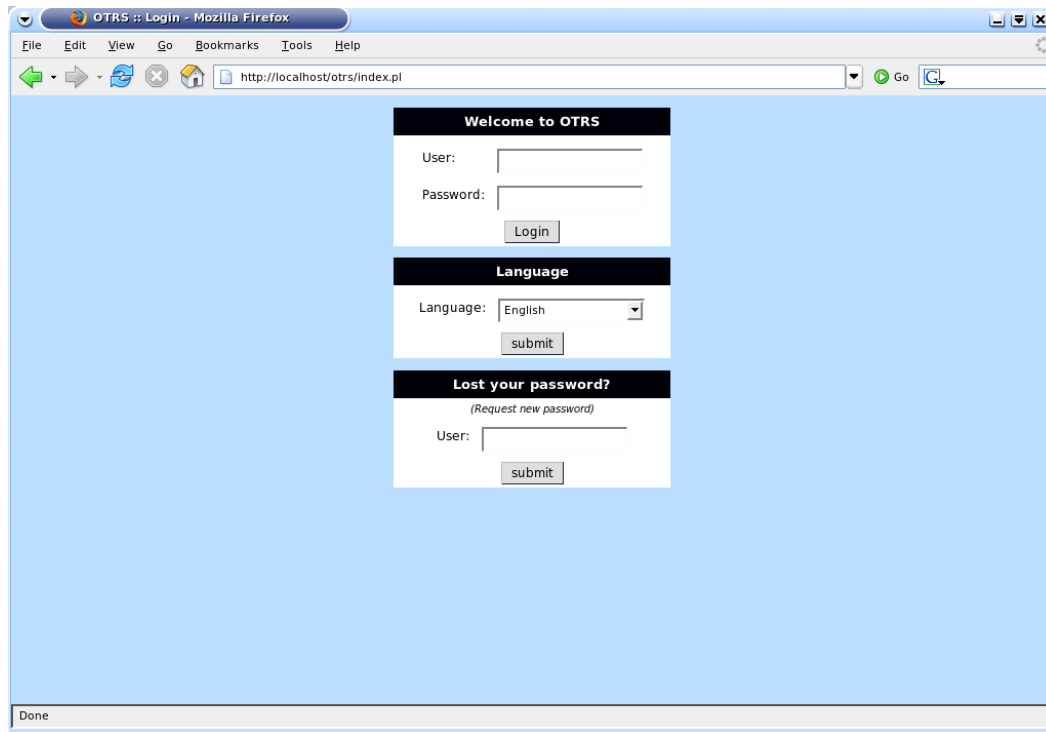
public access to the FAQ system, and lets visitors search through the FAQ entries without any authorization.



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

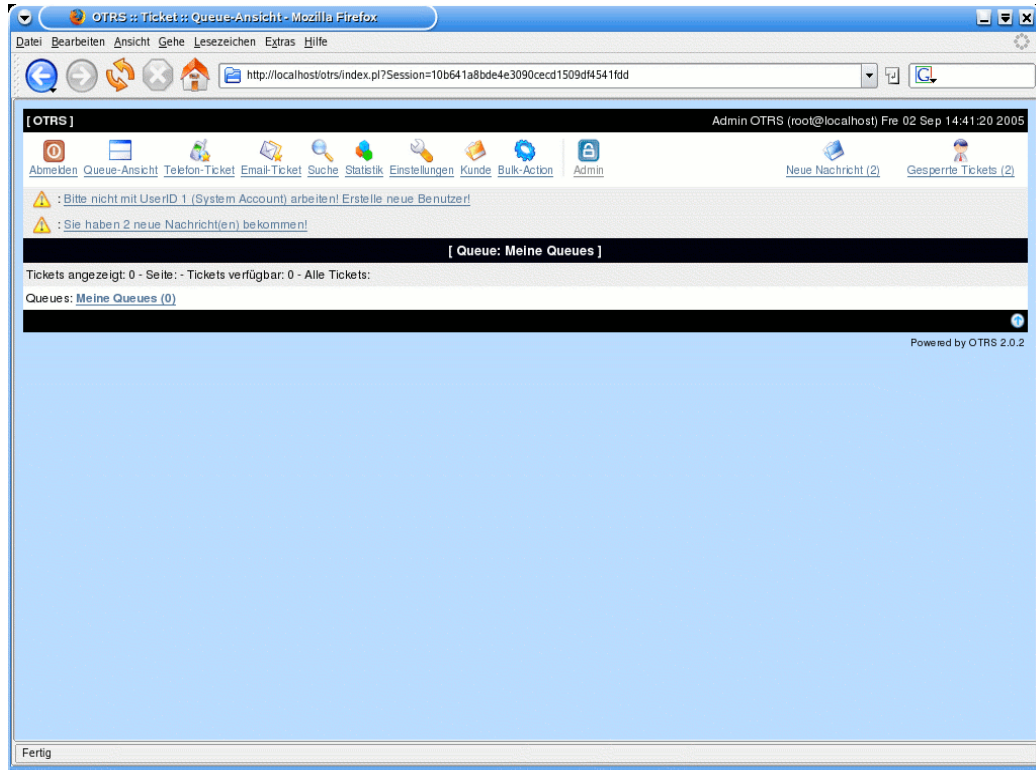
This account data are default on every new installed OTRS system. For that reason you should change the password for the OTRS administrator as fast as possible! This can be done via the preferences for the OTRS administrator account.

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.

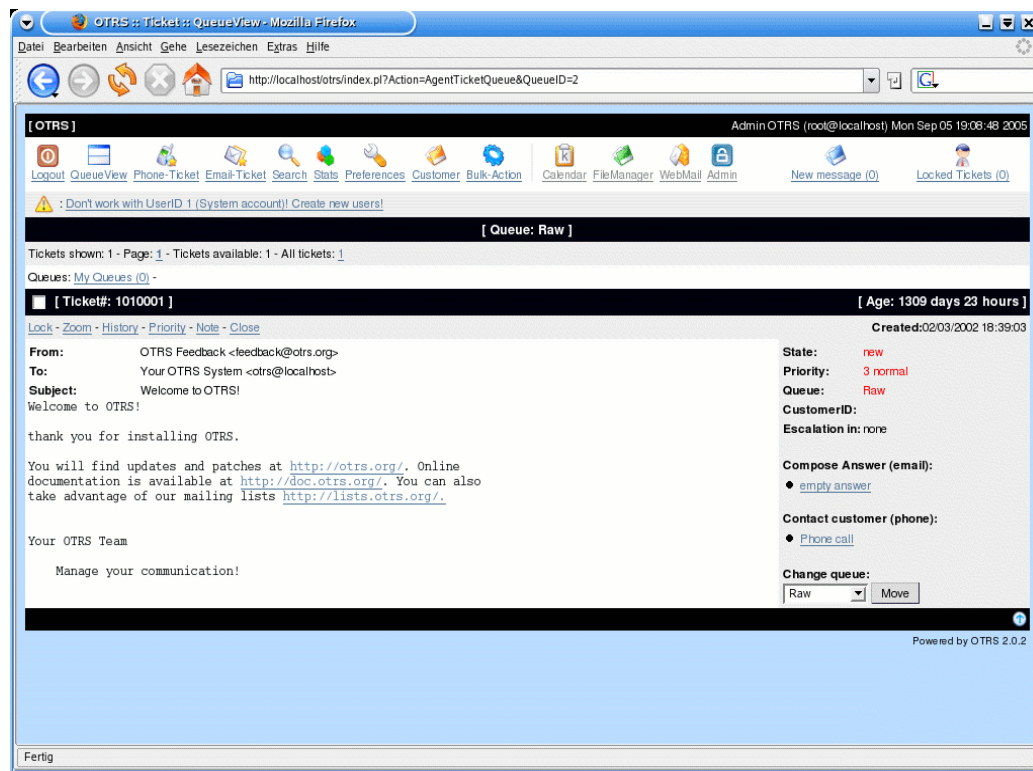
The middle part of the navbar shows the buttons for the navigation to the other modules of the system. After a standard installation you only see the "Admin" button, if you are logged in as OTRS administrator. If some additional application are installed, e.g. the file manager or the web mailer, buttons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

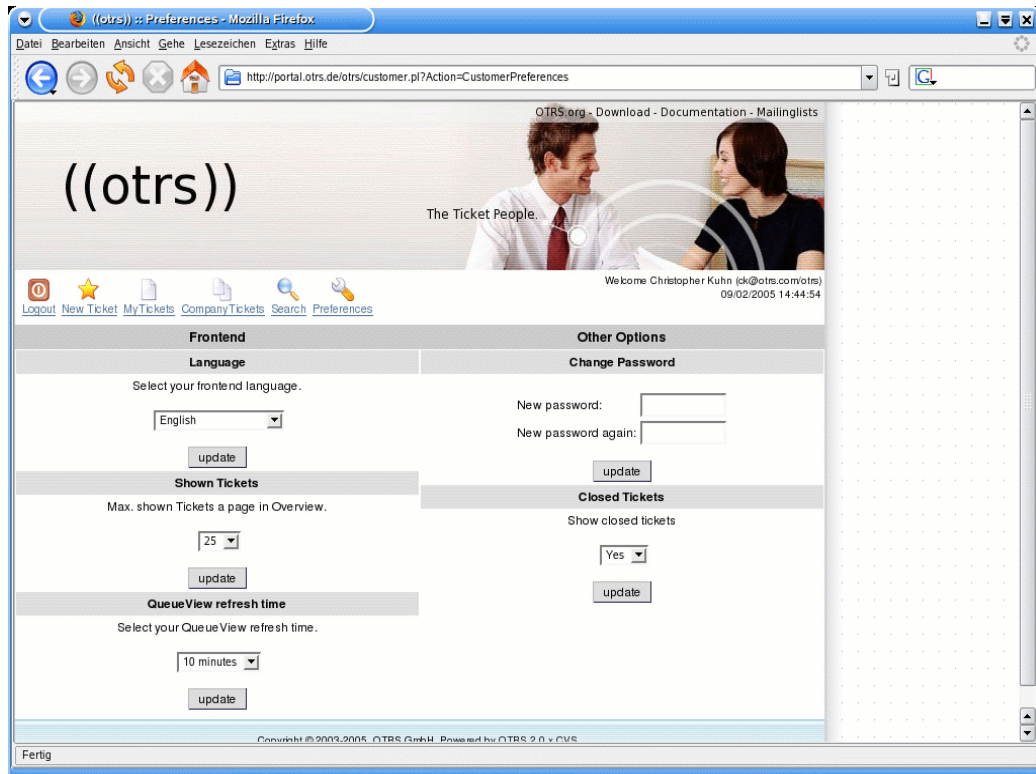
To explain it more practically, remember Max's company describe in the section about a example of a ticket system. Max' has installed OTRS to get rid of his support chaos. He and his agents are using the system to manage the support requests for the video recorders. One queue that contains all request 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.

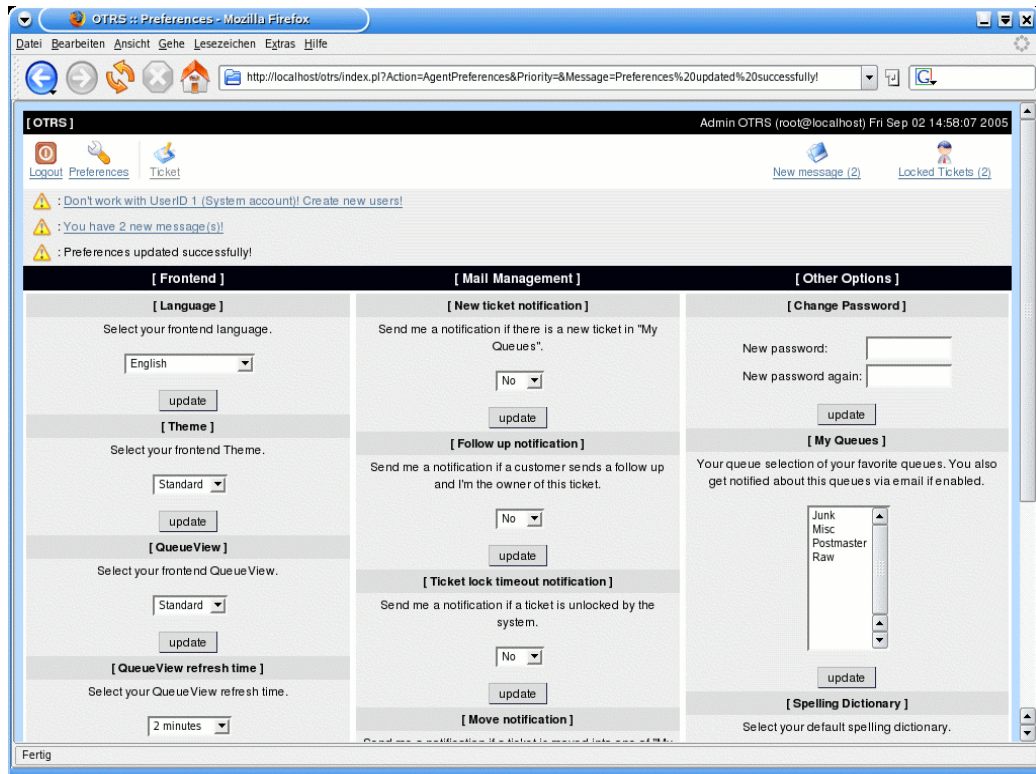
Sorting your incoming messages into different queues helps you to keep the support system structured and tidy. Because your agents are arranged into different groups with different access rights onto queues, the system can be optimized even more. Queues can be used to define work flow processes or they 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

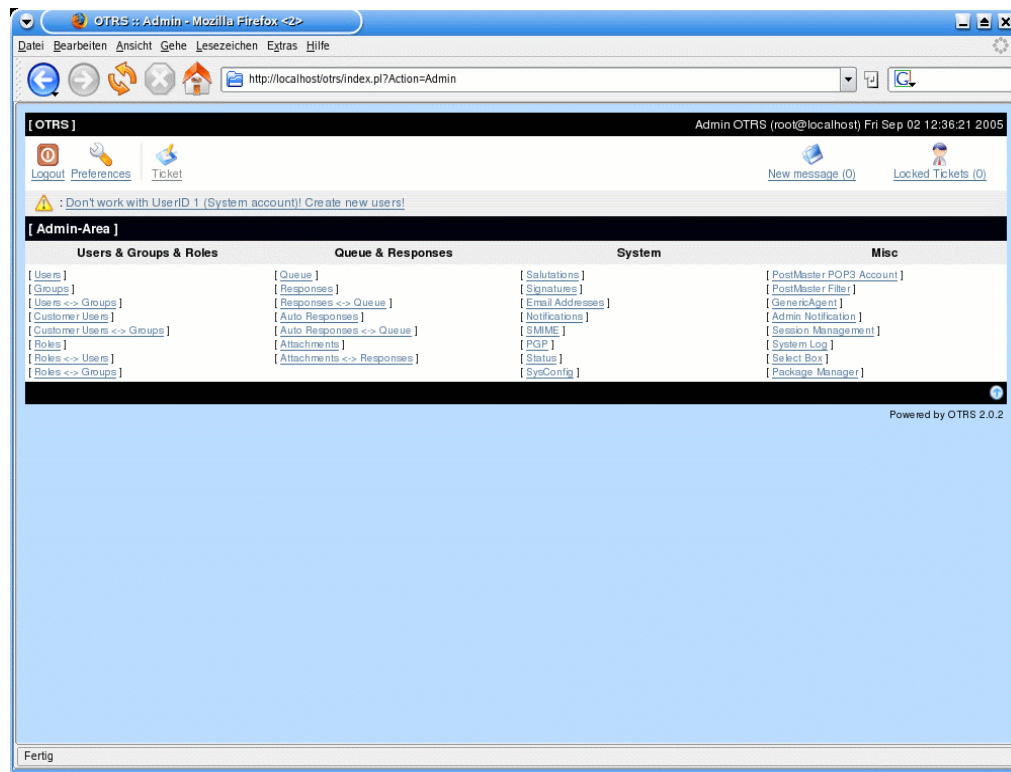
## 5.1. Basics

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.

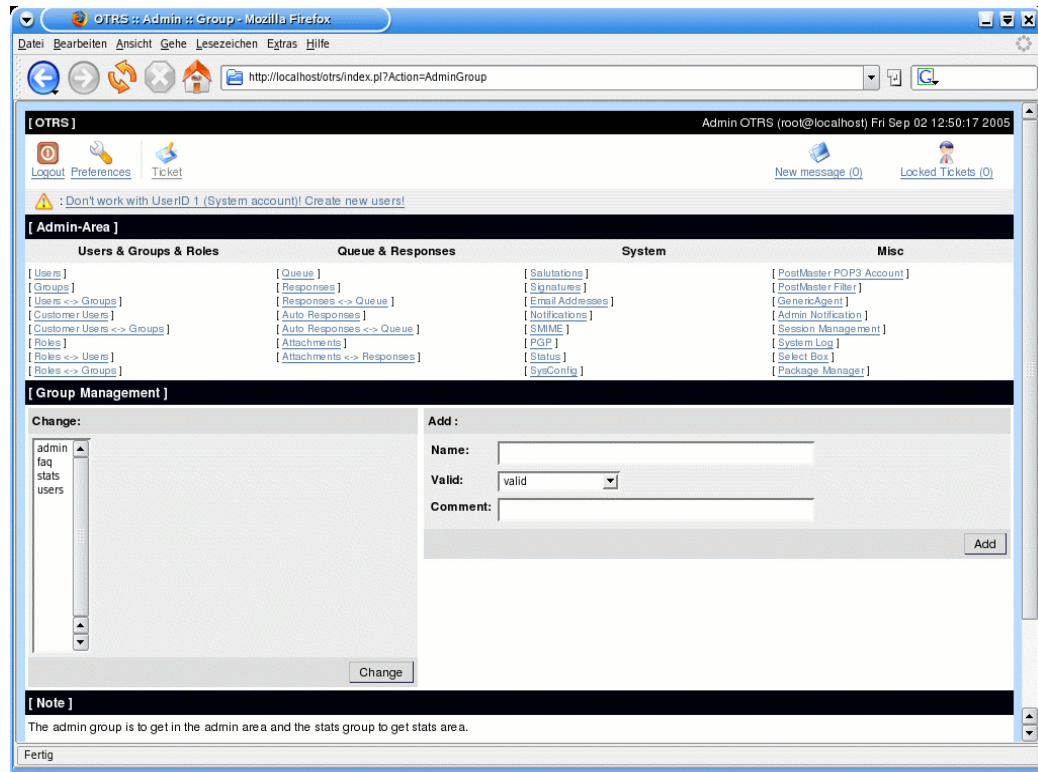
**Note:** To keep the consistency of the data for OTRS it is not possible to delete a user. To deactivate a user account set it to "invalid" or "invalid-temporarily".

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".

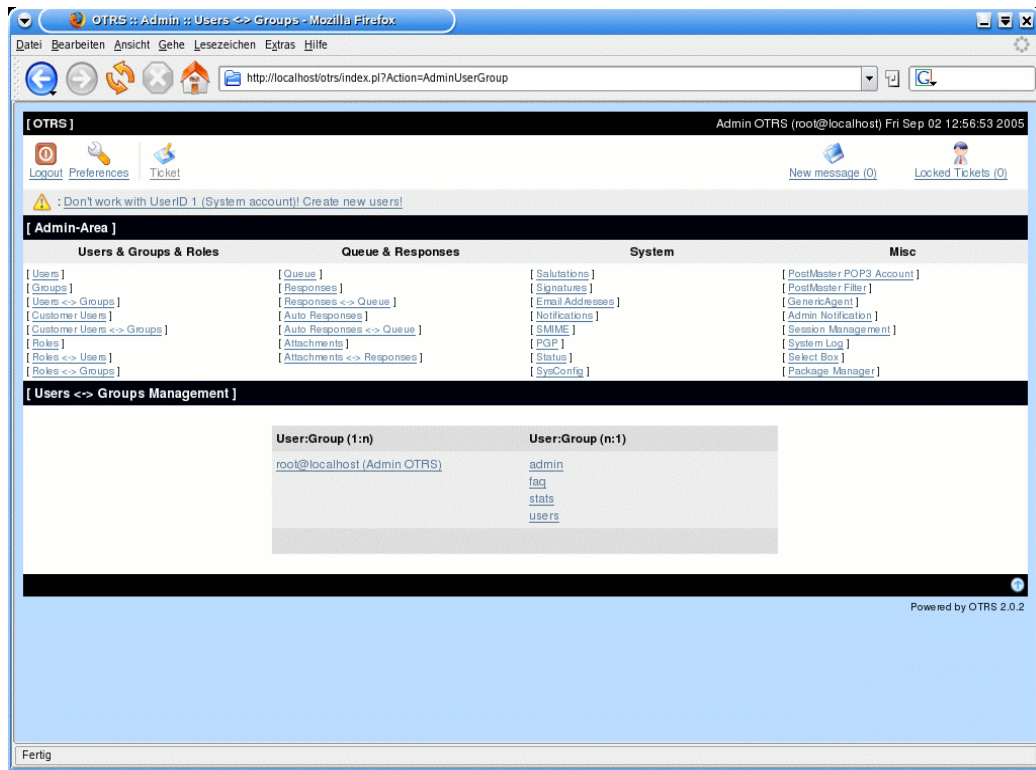
After a default installation already four groups are available in the system.

**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.

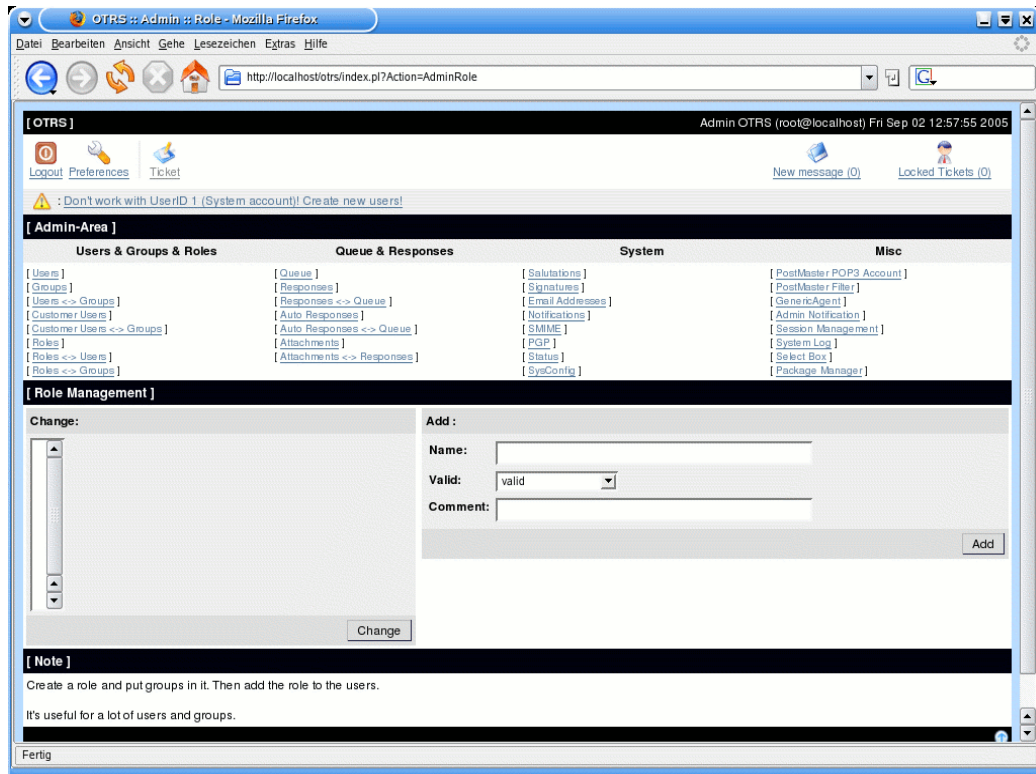
**Table 5-2. Rights in the user groups of OTRS**

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.

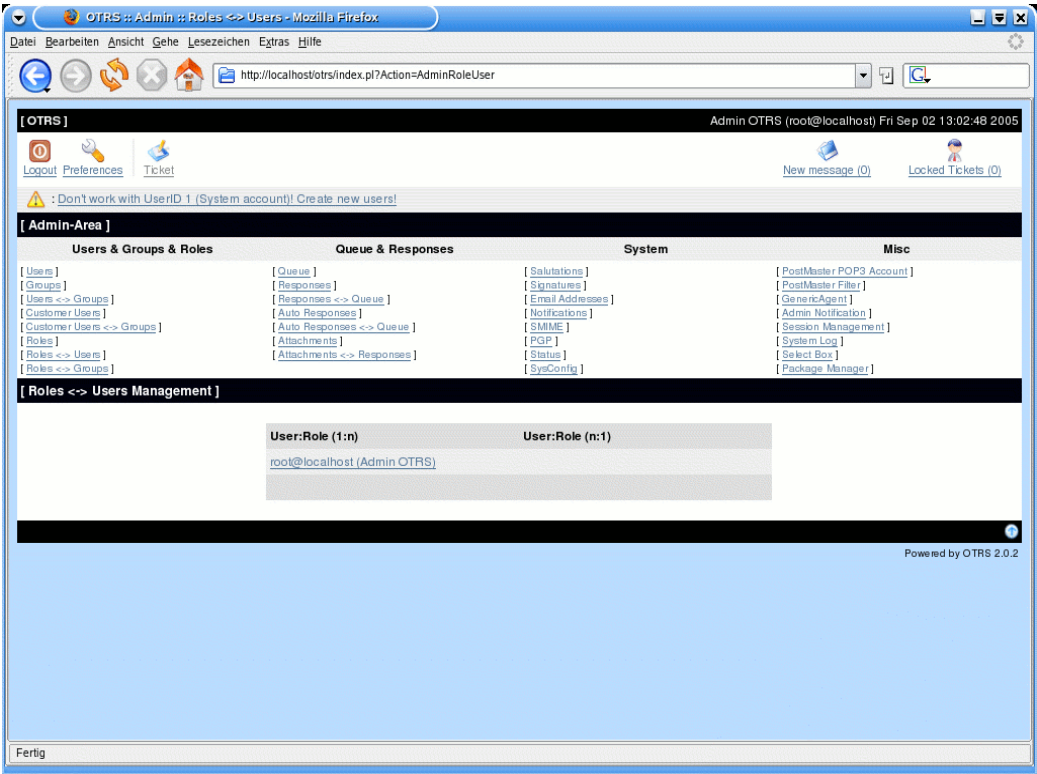
### 5.2.3. Roles

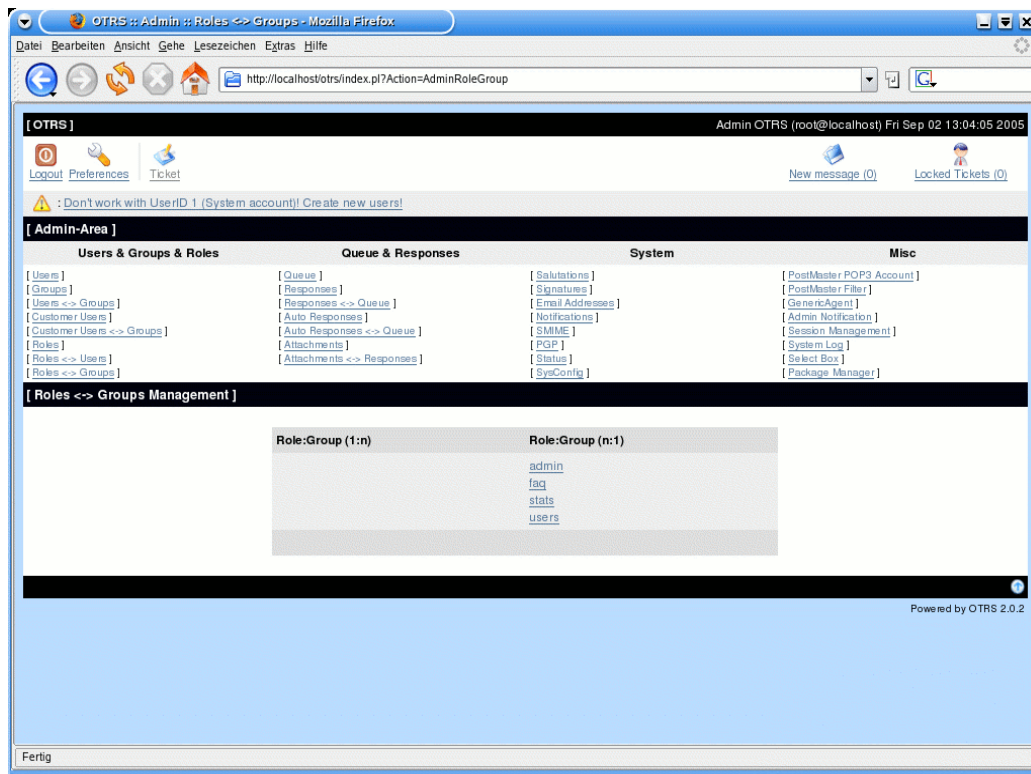
Roles are a very powerful and helpful feature to manage and change the access rights of many users very simply and quickly. On big and complex systems with many users, groups and queues this feature is very useful and helps to save time.

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" bzw. "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. For example the mail address of the user can be specified for Customer#, because a mail address is explicit.

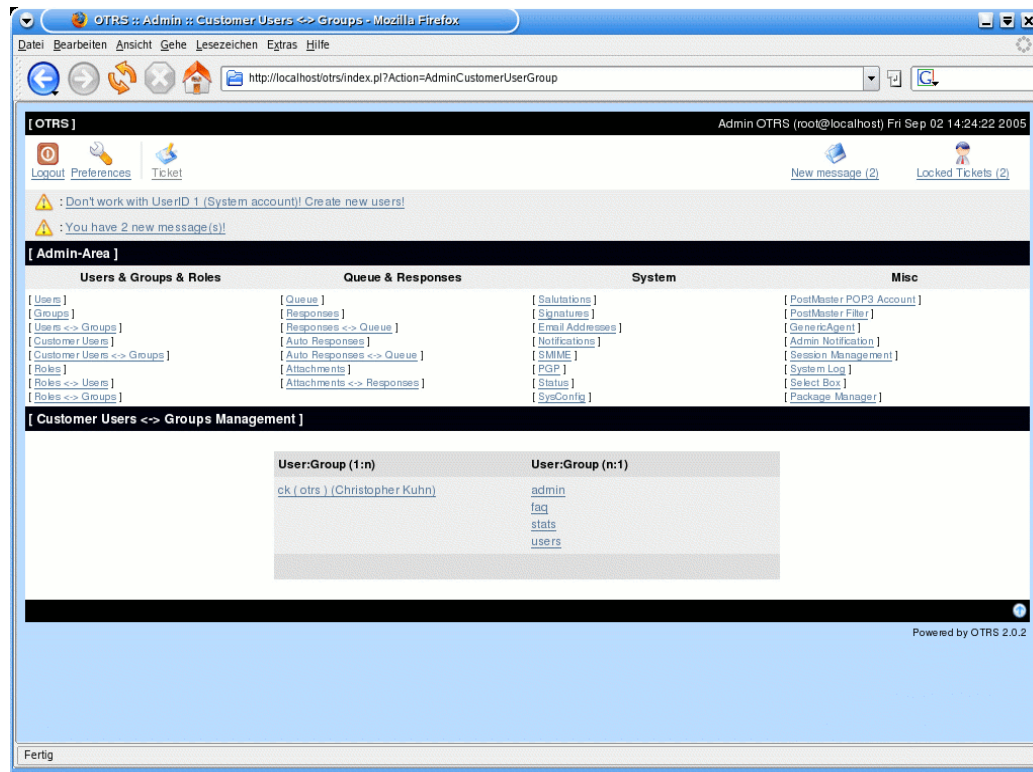
**Note:** To keep the consistency of the data for OTRS it is not possible to delete a customer user. To deactivate a customer user account set it to "invalid" or "invalid-temporarily".

### 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".

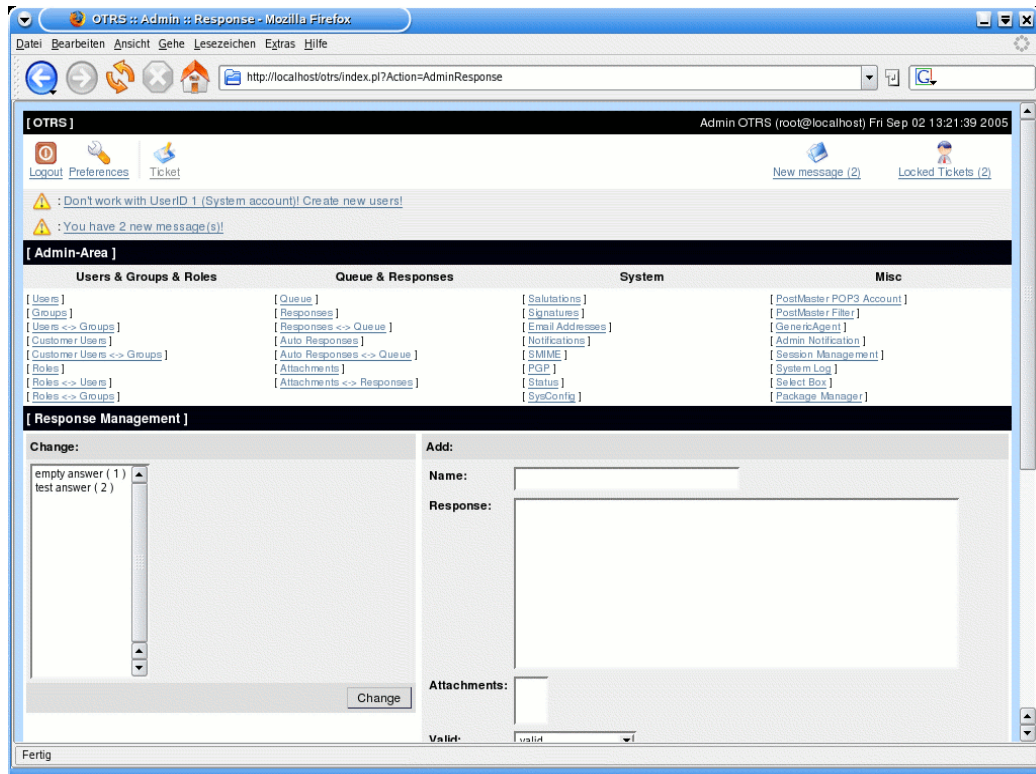
All described config settings for new queues are also valid for sub queues.

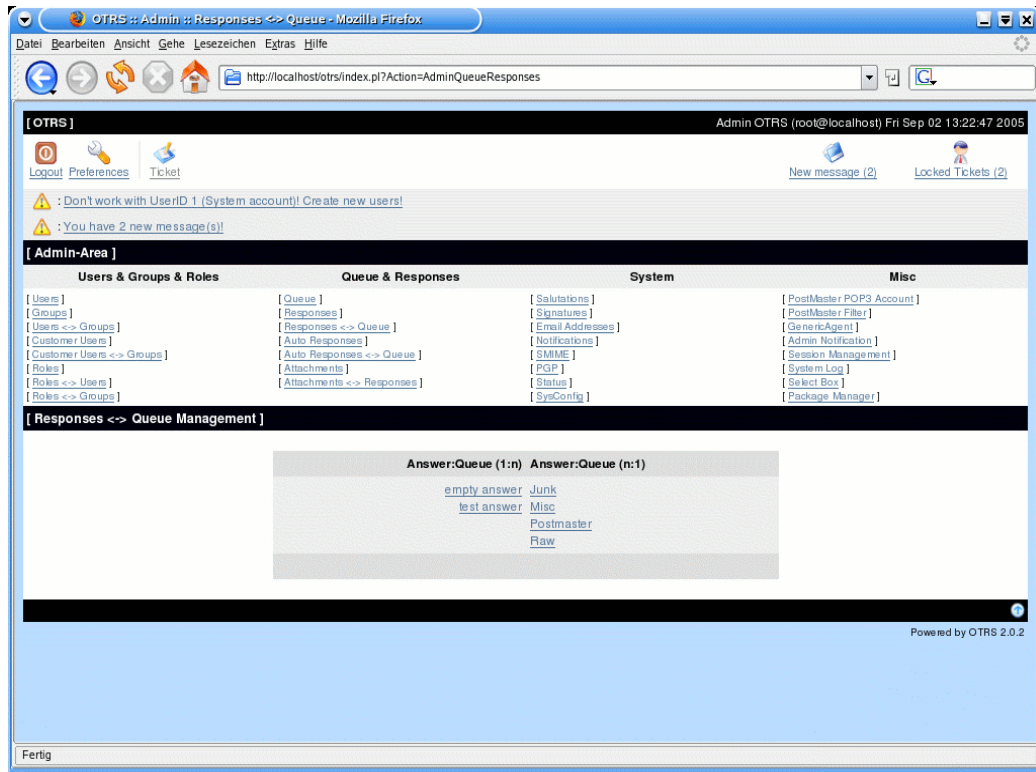
## 5.5. Salutations, signatures, attachments and responses

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.





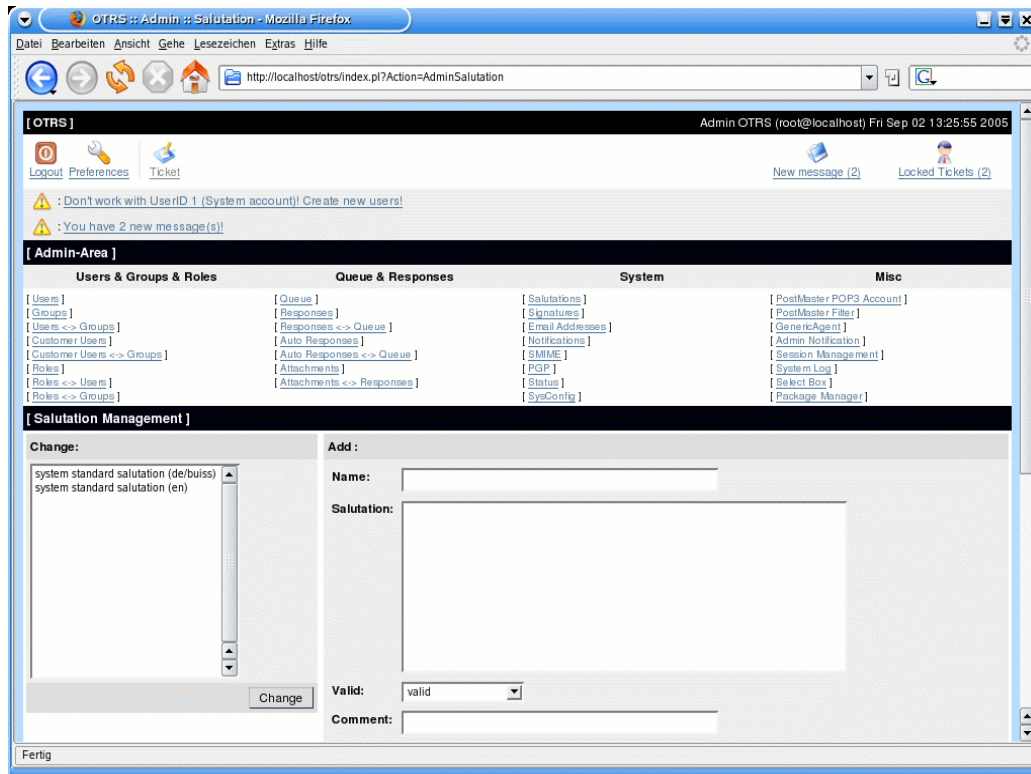


To add or remove response to one or more queues the "Responses <-> Queues" link can be used.

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)".

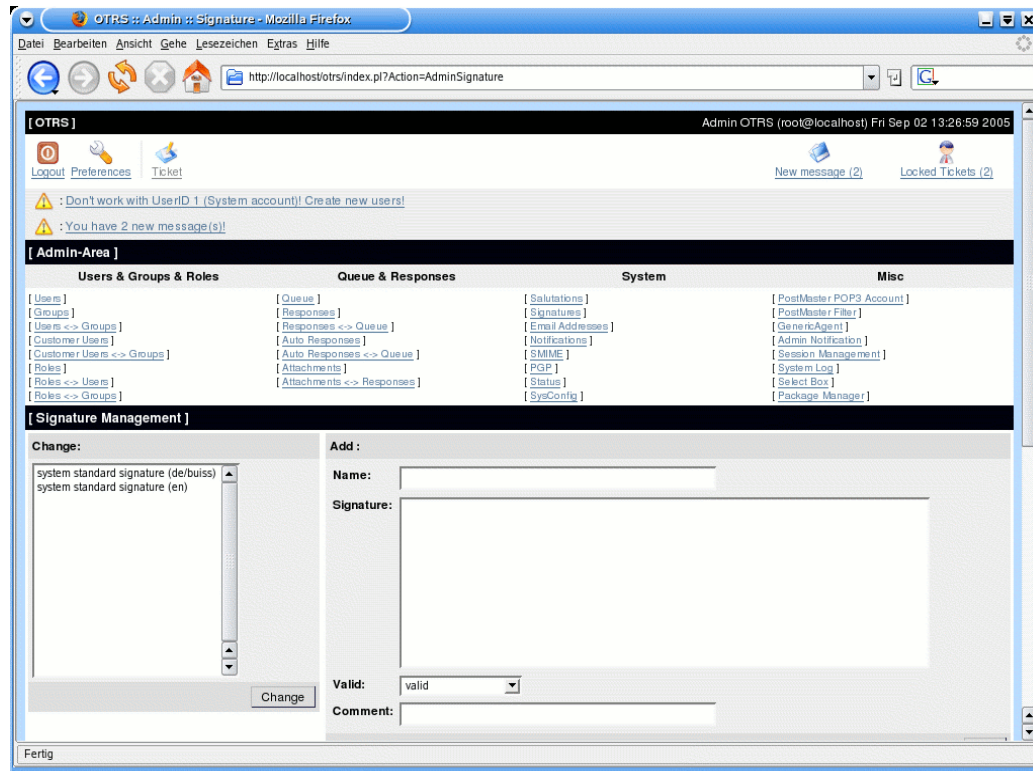
Because the content of a salutation can be created dynamicaly, for example for things that change for every ticket (e.g. the name or mail address of the sender), it is possible to use variables in salutations. The text which is saved in the variables will be placed in the response text if you reply to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



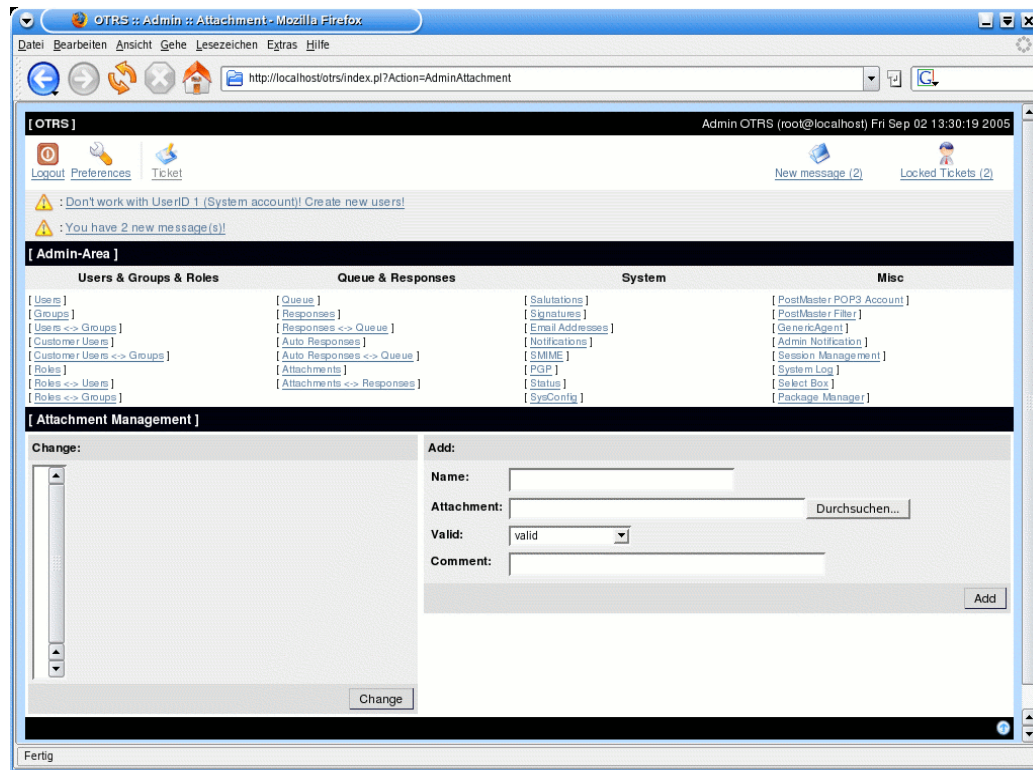
After a default installation of OTRS there are already two signatures stored in your system, "system standard signature (de/buiss)" and "system standard signature (en)".

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.

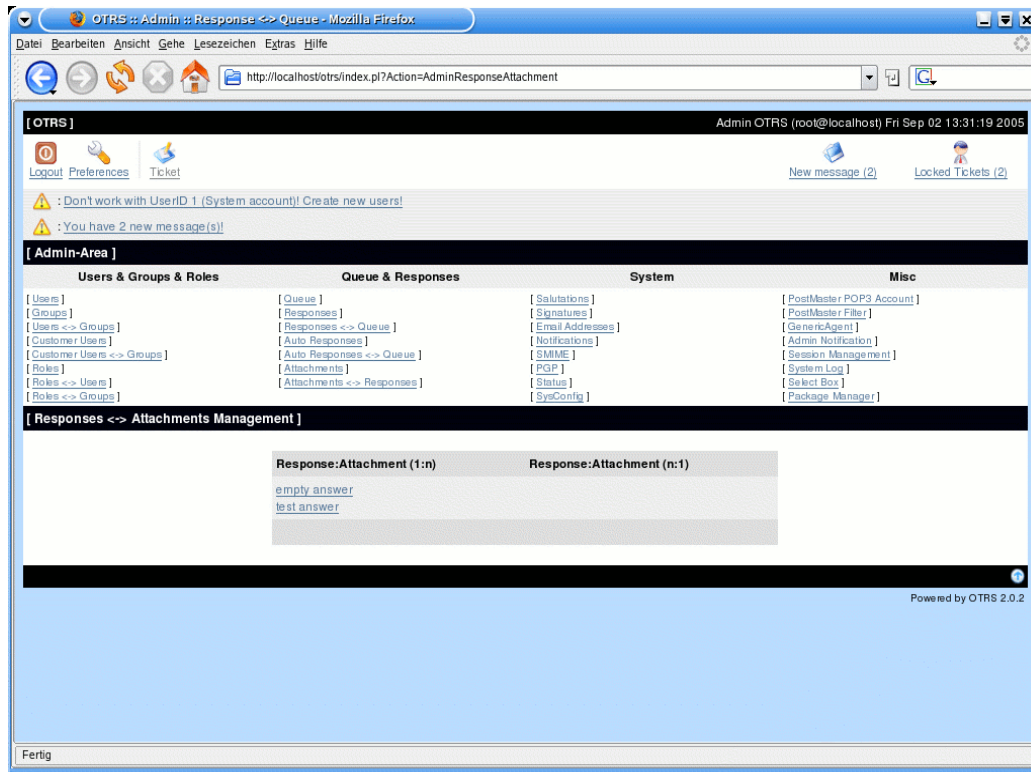
**Note:** To keep the consistency of the data for OTRS it is not possible to delete a signature. To deactivate a signature set it to "invalid" or "invalid-temporarily".

### 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 ticke is receifed, 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.



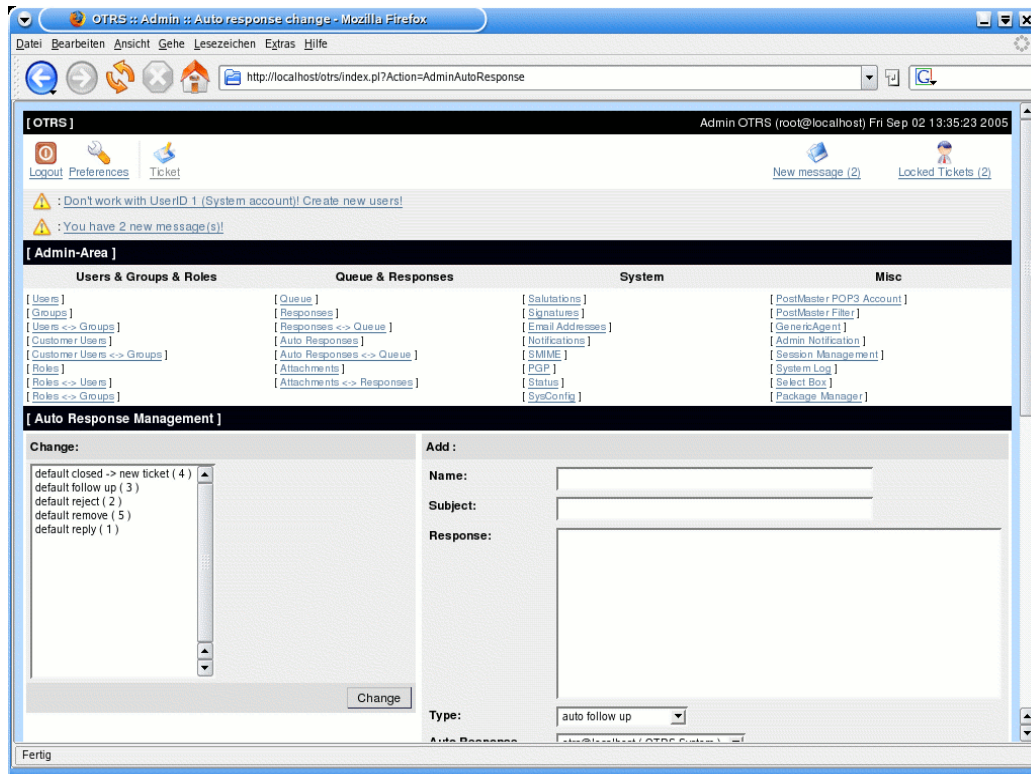
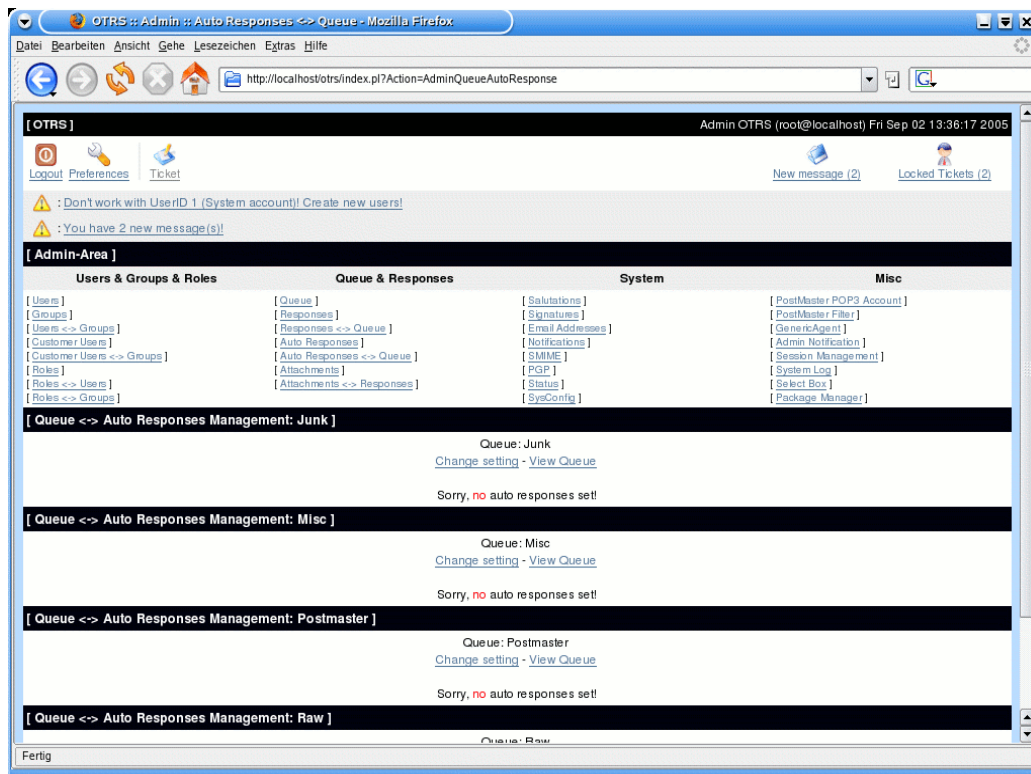


Table 5-3. Events for auto answers

Name	Description
auto reply	This event is triggered when a new ticket is created.
auto reply/new ticket	This event is triggered when a new ticket is created with a new customer reply.
auto follow up	This event is triggered when a new ticket is received by the system.
auto reject	This event is triggered when a ticket is automatically rejected.
auto remove	This event is triggered when a ticket is removed from 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".

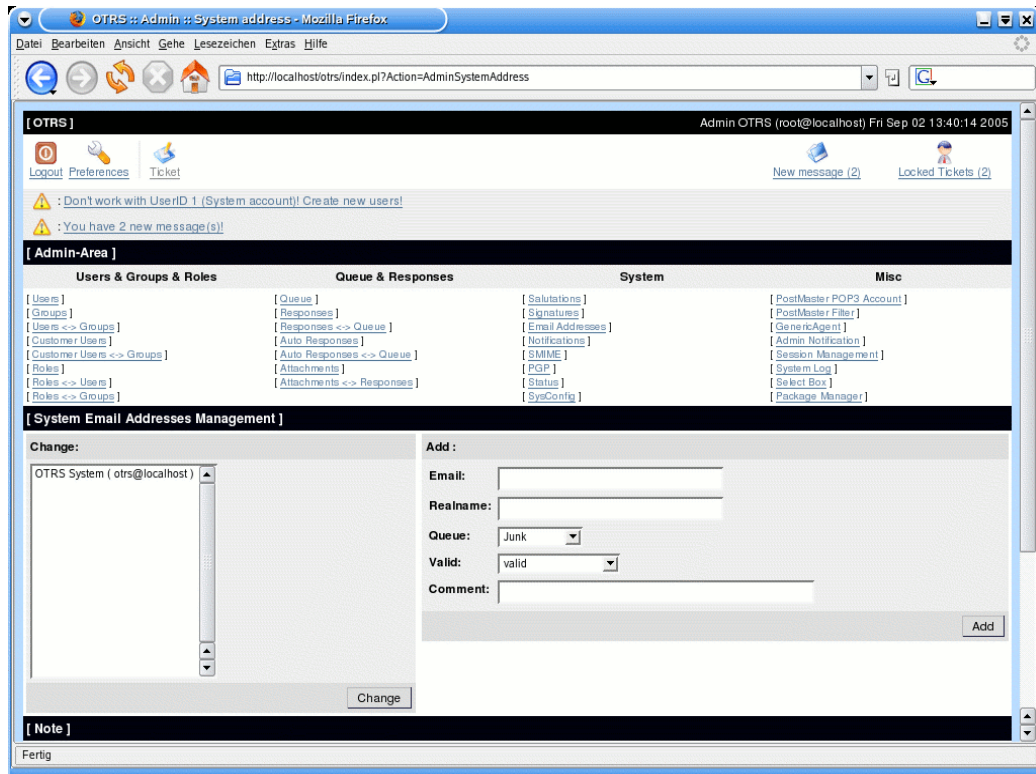


To add an auto answer to a queue follow the "Auto Answer <-> Queues" link in the admin area. All system events are listed for every queue and a auto answer with the same event can be selected or removed via a listbox.

## 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.



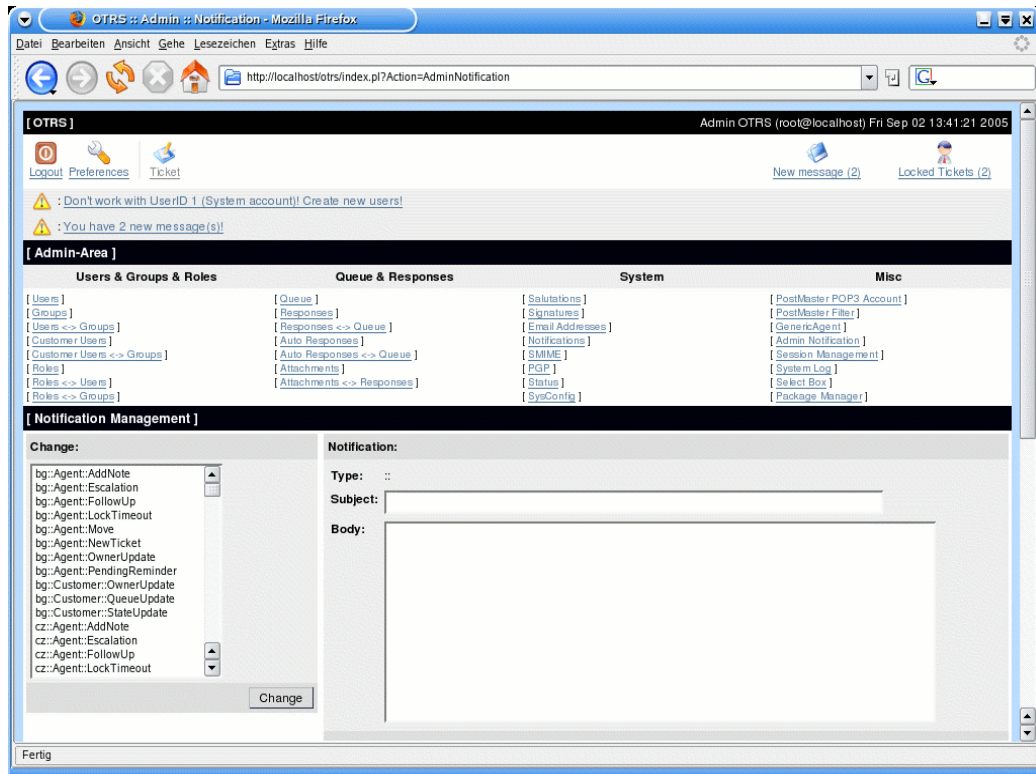


If you create a new mail address you can select the queue or sub queue that shall be linked with the new address. This link enables the system to sort incoming messages via the address in the To: field of the mail into the right queue.

**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.

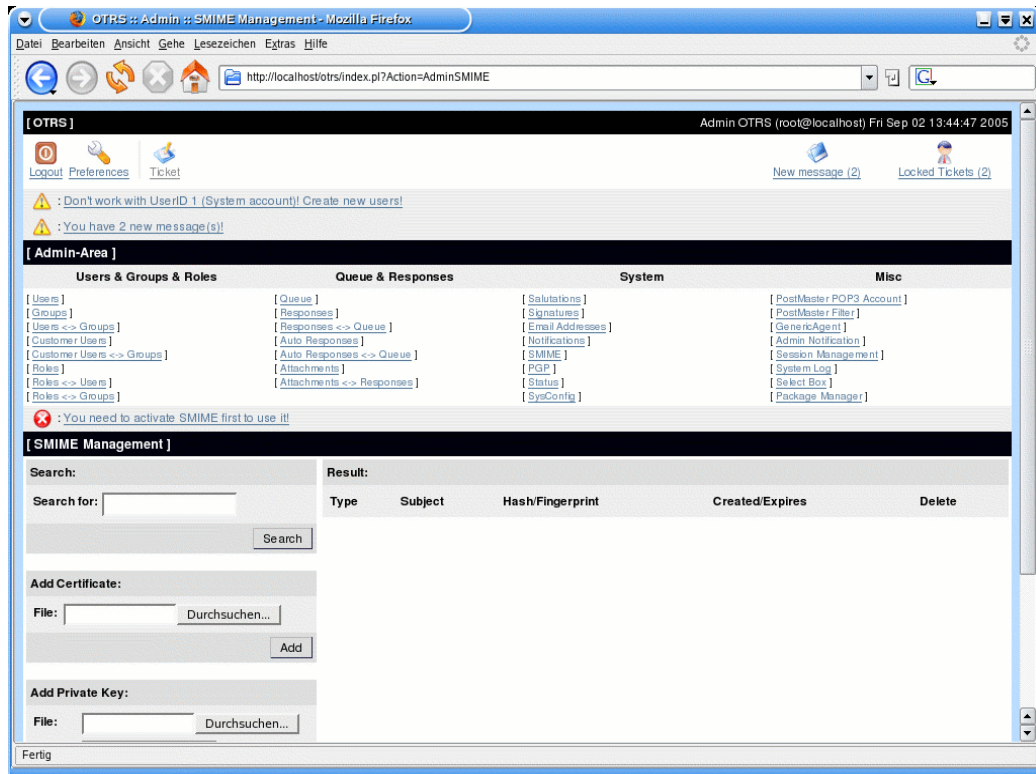


Through the "Notification" link in the adminarea you can manage the notifications of your system. You can customize the subject and the text of the notifications. Just select the notification you want to change from the listbox and load the content of the notification via the "Change" button. The name 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

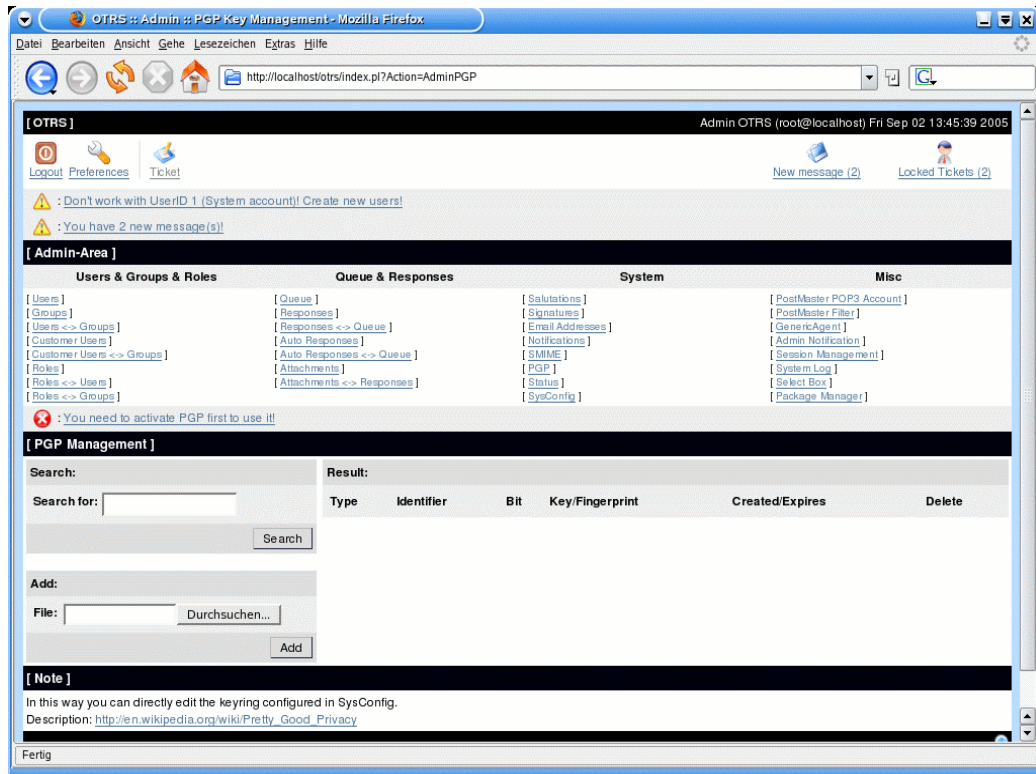
Since OTRS 2.0 it is possible to decode and encode SMIME messages. Also you can sign outgoing mails. Before this feature can be used you need to activate it and change some config parameters e.g. with the graphical configuration frontend to the needs of your operating system.



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

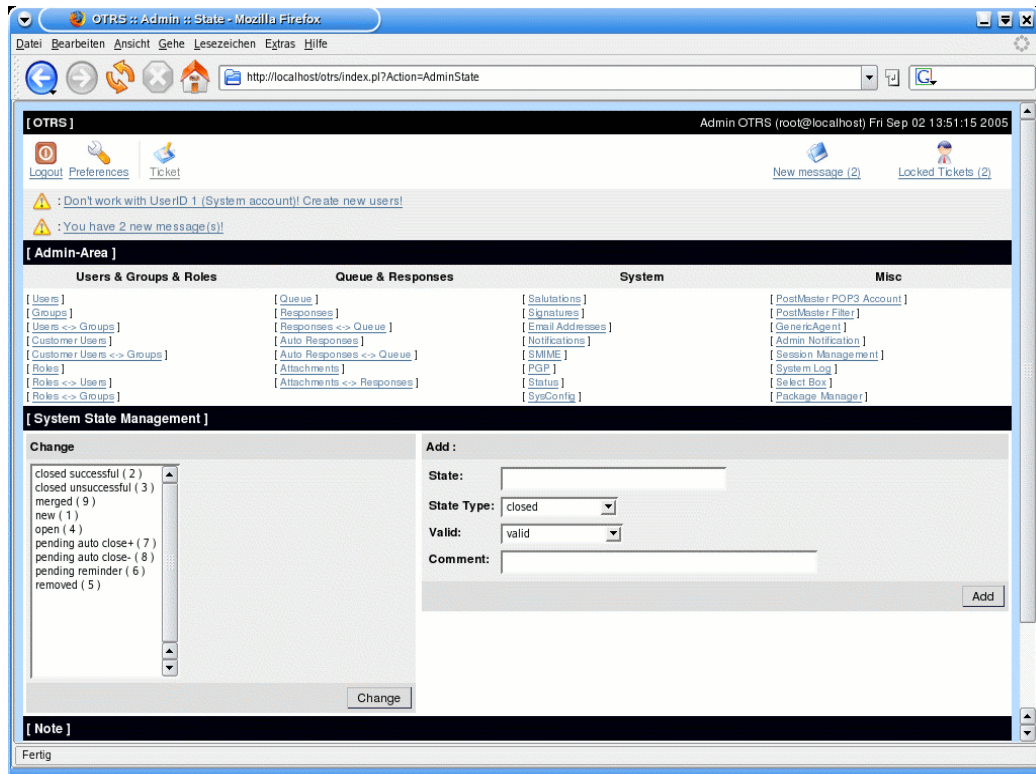
Since OTRS 2.0 it is possible to de- and encrypt messages with PGP. Also you can sign outgoing messages. Before this feature can be used you need to activate it and change some config parameters e.g. with the graphical configuration frontend to the needs of your operating system.



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. Statess

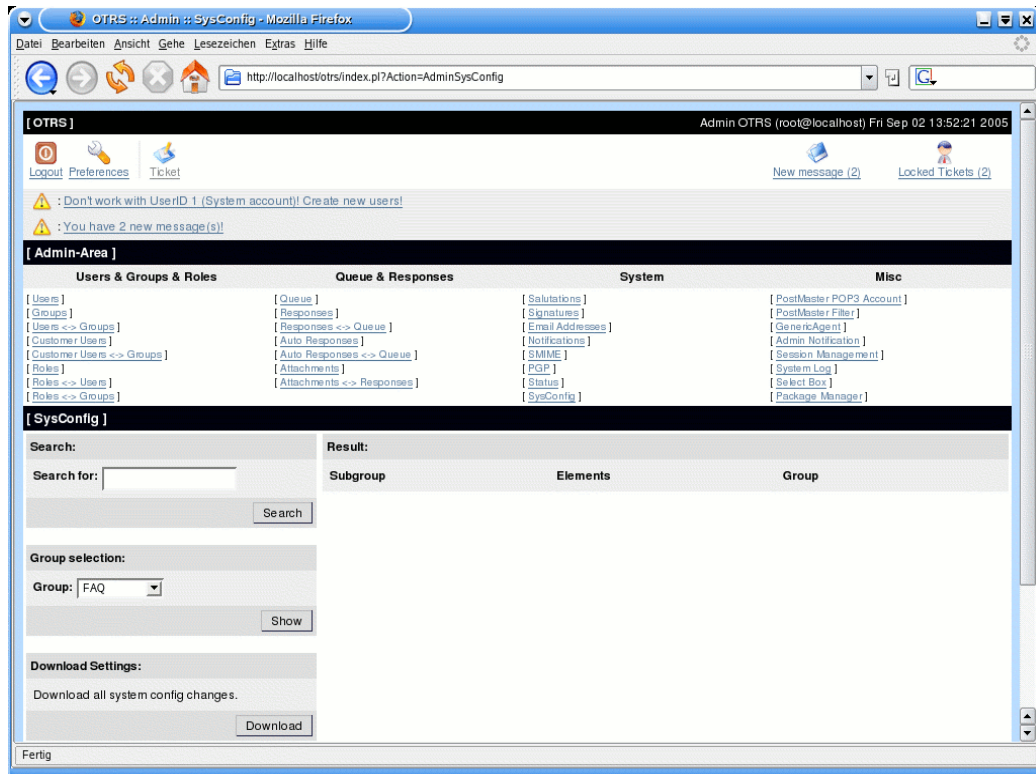
Through the "Status" link in the admin area of OTRS you can manage the different states you want to use in the ticket system.



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.



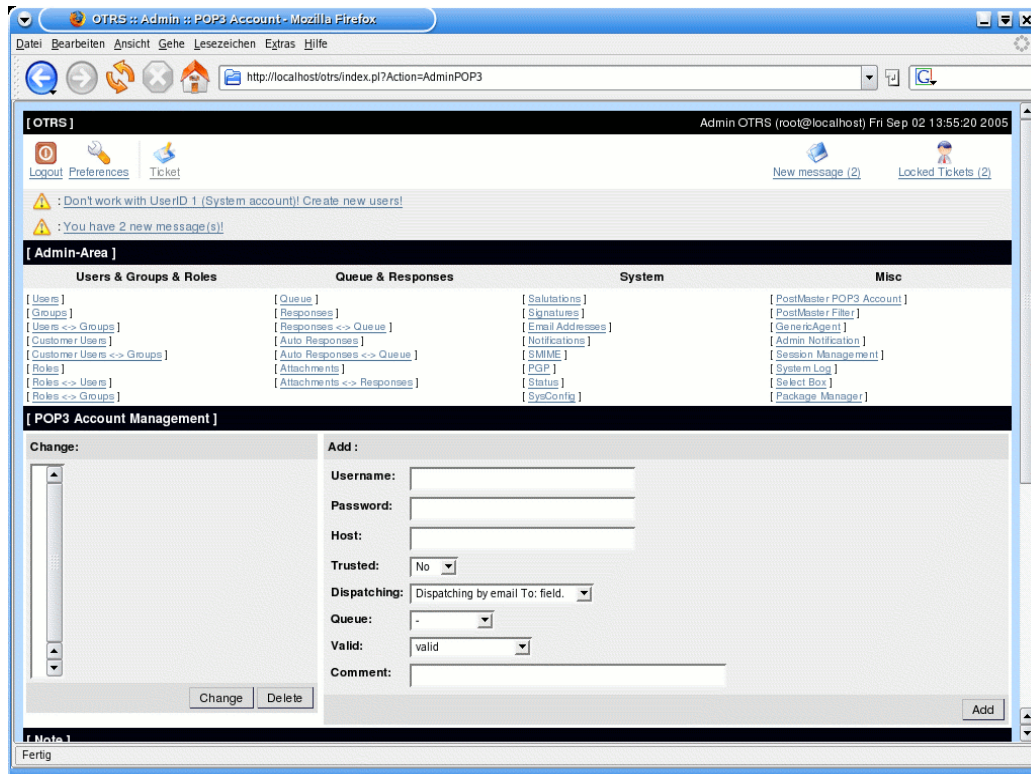
The "SysConfig" link in the admin area of OTRS loads the graphical configuration frontend. You can upload own config files for the system and backup all your current settings into a file. Almost all config parameters of the OTRS framework and the installed applications can be viewed and changed through 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.

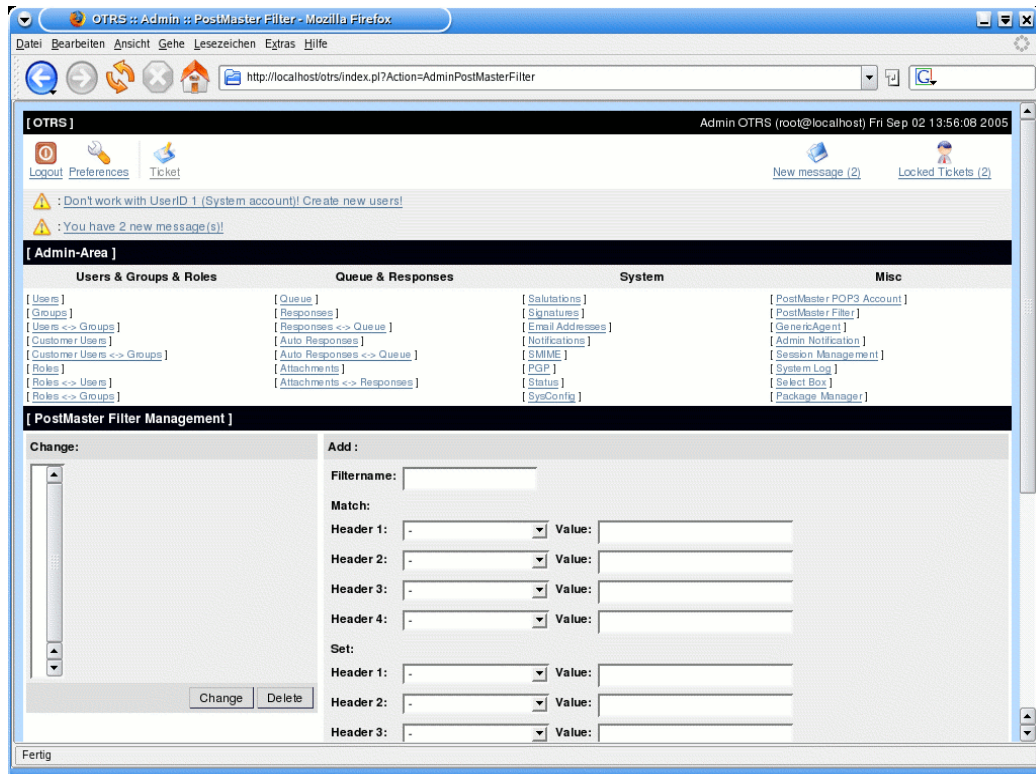




See the section about the PostMaster POP3 accounts for more details.

## 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 bring the 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, after the filter criterias matched. You can define filter criterias for the headers or the body of an email, e.g. search for specific headers 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 matched. The ticket system evaluates the inserted X-OTRS headers and executes the specific actions. X-OTRS headers can be used to sort a 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 a additional info key for this article.
X-OTRS-ArticleValue(1 2 3):	Additional info value for the article.	Saves a 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 a additional info key for the ticket.
X-OTRS-TicketValue(1 2 ... 8):	Additional info value for the ticket.	Saves a 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 oop protection).

A name must be specified for every filter rule. In the section for "Match" the filter criteria can be specified. Chose 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 side the values. In which place in the incoming message you would like to search and specify via the input fields on the right side the values. In the section "Setzen" you set the actions, which should be executed when the filter criteria are met. Here you can choose for "Header 1", "Header 2" etc. the corresponding X-OTRS-Header and enter the corresponding value on the right side.

#### Example 5-1. Aussortierung von Spammails in eine bestimmte Queue

Eine nützliche Filterregel könnte sein, alle Mails, die von spamassassin als Spam markiert wurden,

automatisch in die Queue "Junk" einsortieren zu lassen. Spamassassin fügt bei allen überprüften Mails die Kopfzeile "X-Spam-Flag" hinzu. Wird die Mail als Spam markiert, wird dieser Header auf "Yes" gesetzt. Das Filterkriterium lautet also "X-Spam-Flag: Yes". Um nun eine Filterregel mit diesem Kriterium zu erzeugen, tragen Sie hierzu als Name für die Filterregel z.B. "spam-mails" ein und wählen in der Sektion "Treffer" für "Header 1" den Header "X-Spam-Flag:" aus der Listbox aus. Rechts daneben fügen Sie als Wert "Yes" hinzu. Das Filterkriterium wurde somit festgelegt. Um nun die Einsortierung der von spamassassin als Spam klassifizierten Mails in die Queue "Junk" zu erzwingen, wählen Sie in der Sektion "Setzen" für "Header 1" den Eintrag "X-OTRS-Queue:" aus und tragen als Wert rechts daneben "Junk" ein. Abschließend wird mit Hilfe des "Hinzufügen"-Buttons die neue Regel gespeichert und aktiviert, beim nächsten Abruf der POP3-Konten bzw. für die nächste an das System gesendete Nachricht wird die Filterregel abgearbeitet.

Es gibt weitere Module, die zum Filtern eingehender Emails genutzt werden und bei komplexeren Installationen nützlich sein können.

## 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

[ Roles <-> Groups ] [ SysConfig ] [ Package Manager ]

[ GenericAgent ]

Job-List

**Save Job as?**

Name:

**Is Job Valid?**

Is Job Valid: Yes ▼

**Schedule**

minutes: 50 hours: 05 days: Fri

**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**

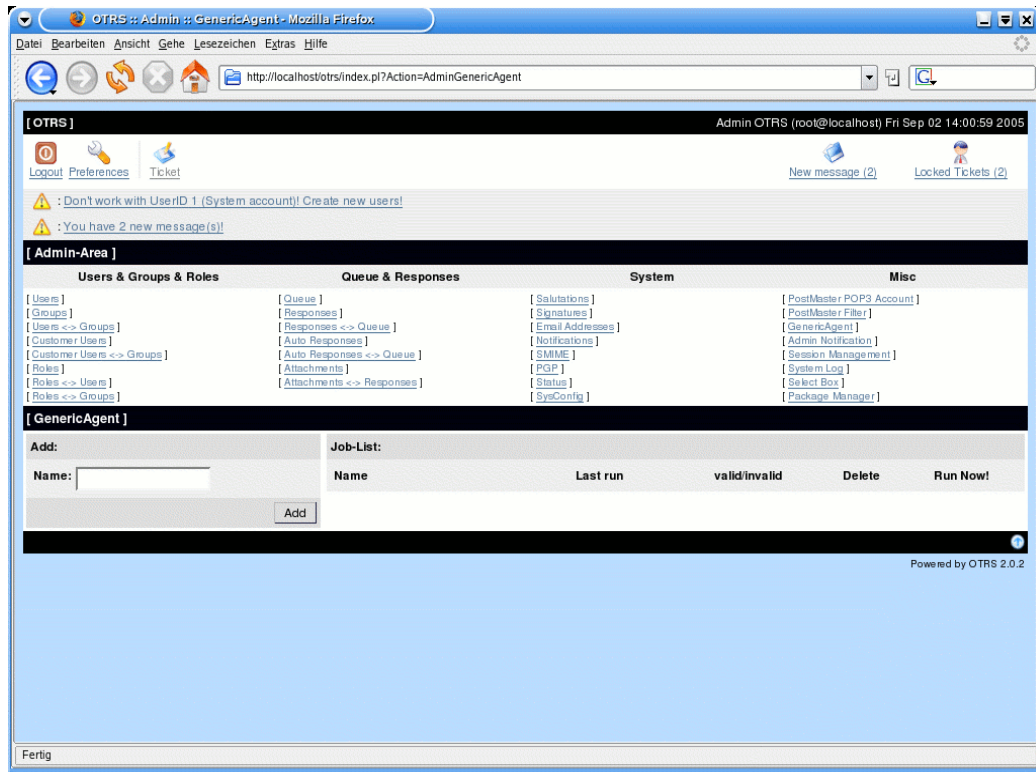
closed successful  
closed unsuccessful  
merged  
new

**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.

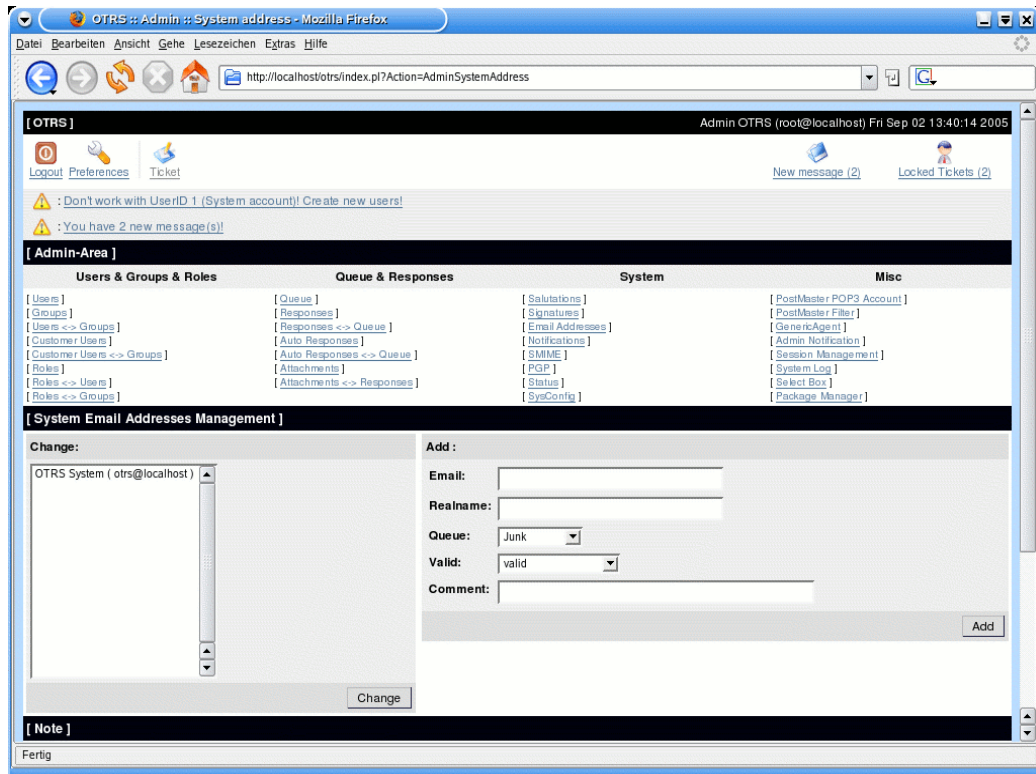


Via the screen for the creation of a new job for the GenericAgent the times can be specified when the job shall be executed. Also different criteria can be specified to select the tickets that shall be affected by the job. Also it is possible to set the new properties of the tickets which are affected 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 lets you controll if the job works right, no changes are made to this tickets yet. The job will be really activated if it is saved into the job list.

## 5.16. Admin email

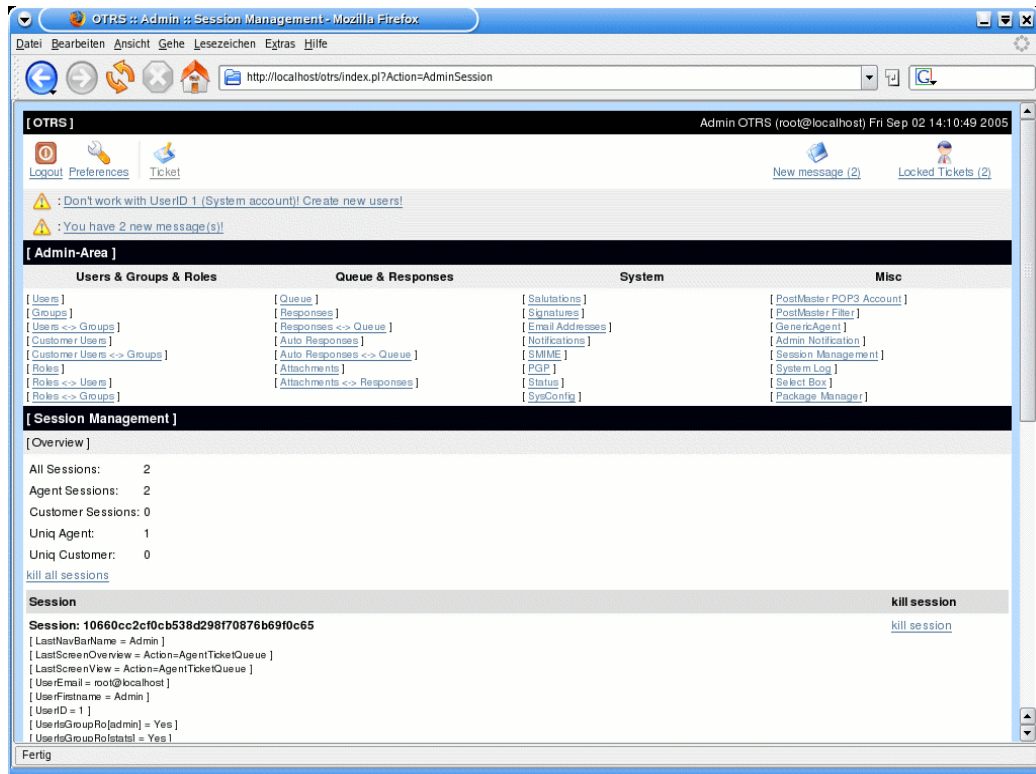
The OTRS administrator can send a message to specific users or user groups. The "Admin Notification" link opens the screen where the users and groups can be selected that should be notified by the admin.



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

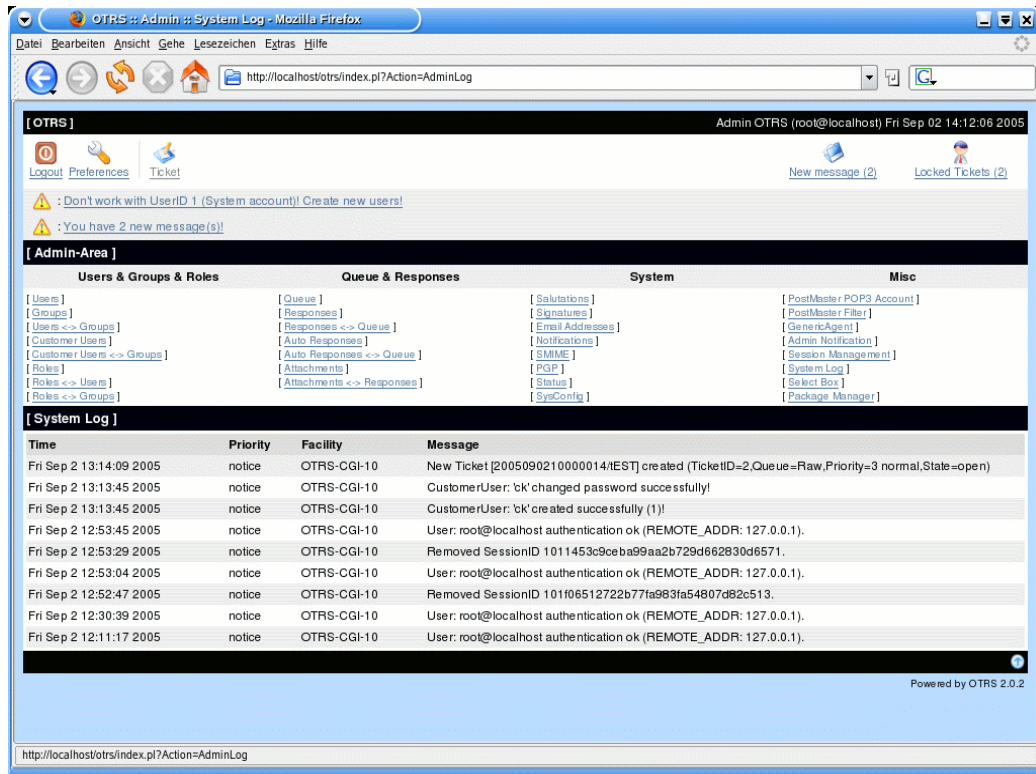
To get an overview on the currently logged in users and their session properties the "Session Management" link in the admin area can be used.



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.

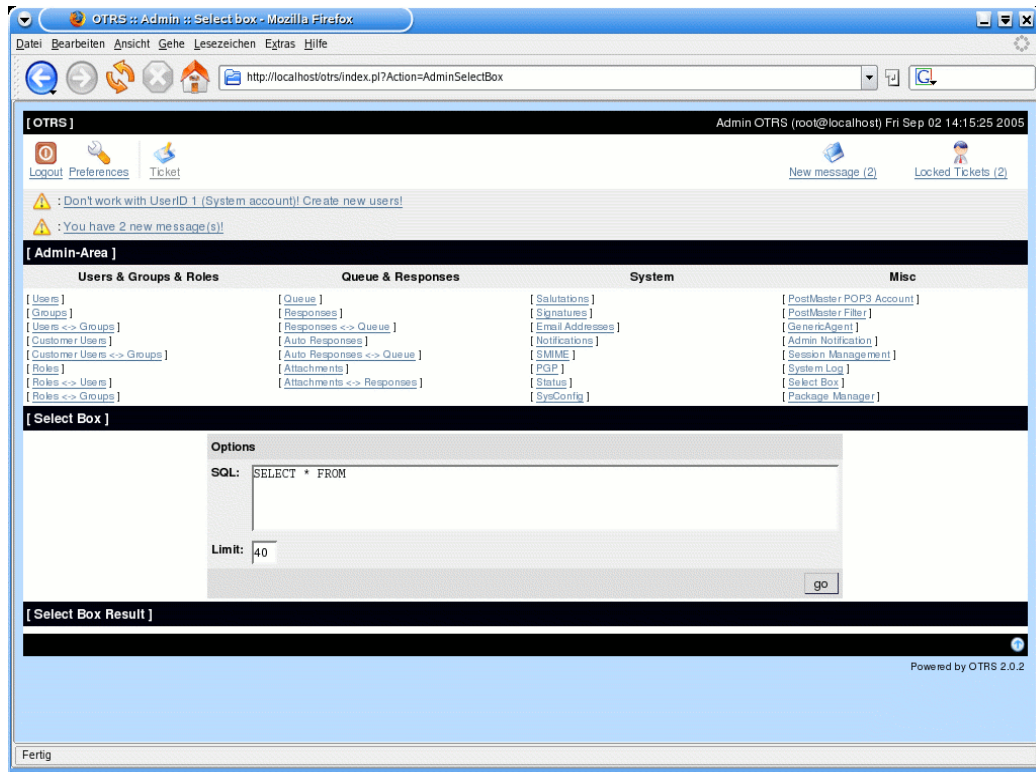
**Note:** The system logs are only available via the web interface on linux or unix systems.

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.

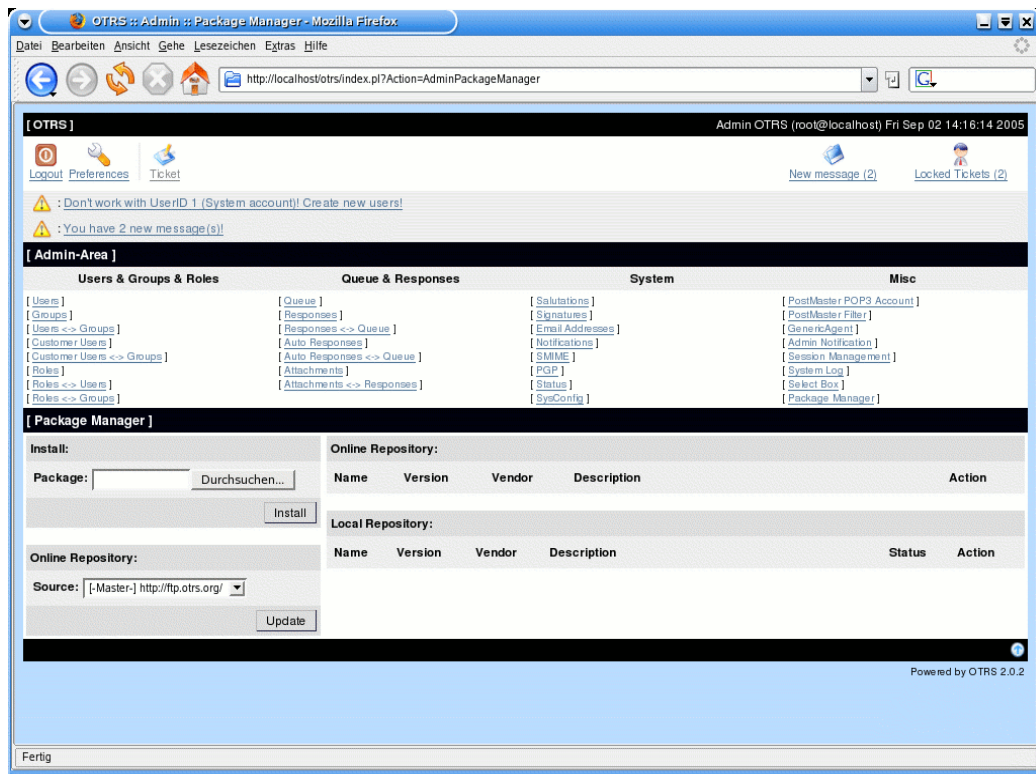




## 5.20. Package manager

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 newest 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. Dependences 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 other files than `Kernel/Config.pm` manually because 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 save 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.

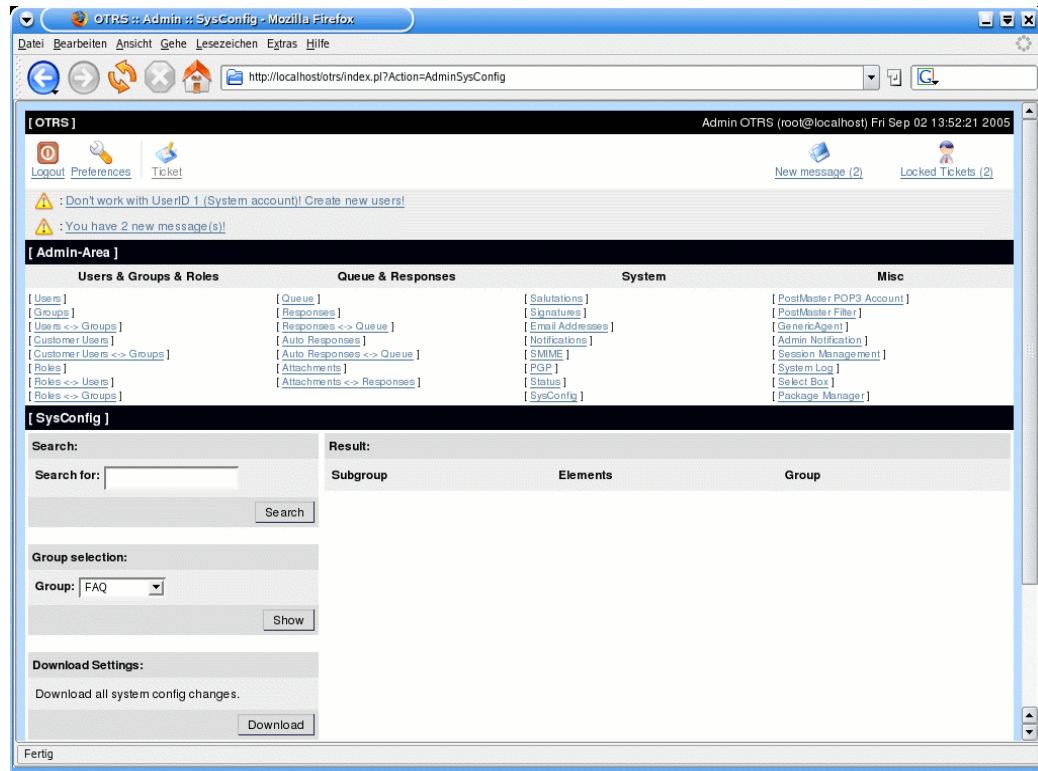
In the directory `Kernel/Config/Files` some more files are stored that are parsed when the OTRS loginpage is accessed. If additional applications like the web mailer or the file manager are installed, the configuration files of these applications can be found in `Kernel/Config/Files`. Because of compatibility reasons to older OTRS versions, at present there is always a `.pm`- and a `.xml` file for every application and for the central framework available. 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 since OTRS 2.0 and which lets you configure most of all configuration parameters of the system through your web browser. In future versions of OTRS the `.pm` files will be removed and the system will 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. At last the file `Kernel/Config.pm` will be read out that contains your individual settings and manually changed configuration parameters. Reading out 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

Since OTRS 2.0 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 has more than 600 configuration parameters at this time, there exist different ways to access a specific configuration parameter quickly. 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, also the descriptions of the parameters are scanned. 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 no longer take care on this parameter or the default is used. With the "Reset"-button it is possible to switch a changed configuration parameter back to the system default. 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 system parameters which 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 can not 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 out emails via SMTP ( Simple Mail Transfer Protocol / RFC 821 (<http://www.ietf.org/rfc/rfc821.txt>)). This settings will mostly be needed 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`:

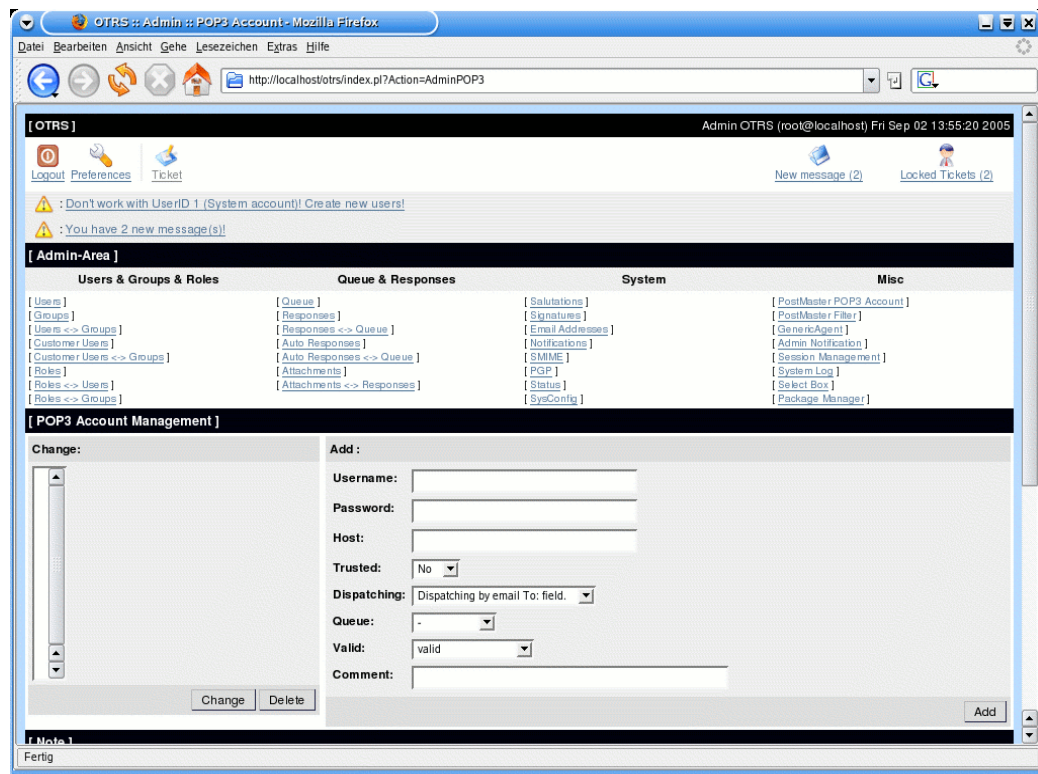
```
# SendmailModule
$Self->{"SendmailModule"} = "Kernel::System::Email::SMTP";
$Self->{"SendmailModule::Host"} = "mail.example.com";
$Self->{"SendmailModule::AuthUser"} = "";
$Self->{"SendmailModule::AuthPassword"} = "";
```

## 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 a POP3 server, a 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 "Trusted" only set to "Yes" for such senders that are known very well. X-OTRS-Header are used by the filter module in OTRS. The X-OTRS headers are explained in this table more detailed. If you have created filter rules they are executed and used even if "Trusted" is set to "Yes".

Also the distribution of incoming messages can be controlled if they shall 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 queue that can be specified also. 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. Is the address in the To field linked with a queue, the new message will be sorted into the linked queue. Can no link between the address in the To: field and queue be found, 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 database of OTRS. The PostMasterPOP3.pl, script which is located in the bin directory of your OTRS installation, uses the settings in the database and fetches the mails. You can execute `./bin/PostMasterPOP3.pl` manually to check if all your POP3 settings are working properly. If you want to fetch your mails automaticly 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 emails into OTRS, the commandline programm `bin/PostMaster.pl` might be a solution. `bin/PostMaster.pl` takes the mails via STDIN and pipes them directly into OTRS. That means emails 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 on the command line 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 works fine.

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 emails 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 e-mails from your mail server via a POP3 or IMAP mailbox to the OTRS machine/local otrs account and to procmail use fetchmail (<http://www.tuxedo.org/~esr/fetchmail/>).

**Note:** A working SMTP configuration on the OTRS machine is 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 e-mails will be forwarded to the local otrs account, if the command **fetchmail -a** is executed. Setup 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 exist 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',
    },
};
```

Ofcourse it's also possible to develop own PostMaster filter modules.

# Chapter 8. Using external backends

## 8.1. Customer data

OTRS can handle different customer data, e.g. login, mail address, phone number. This information can be displayed, edited and used via the agent interface. Also the customer need these data to authenticate to the ticket system.

The customer data which will be displayed or used in the system is freely configurable, but OTRS needs at least three informations about a customer to work properly, the login for the customer, the mail address of the customer and the customer ID.

If you want to display the customer data (e.g. company, name, email) in the agent interface use the following config parameters and insert them in `Kernel/Config.pm`

```
# 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;
```

## 8.2. Customer user backend

OTRS can use two types of customer user backends, DB and LDAP. If you already have a database or directory with customer data (e.g. SAP) it is ofcourse possible to write a new backend.

**Note:** It is also possible to configure more than one customer data backend for OTRS. Just add another config option like "CustomerUser" with a number between 1 and 10, e.g. "CustomerUser1".

### 8.2.1. Database (Default)

This is an example for a database customer user backend.

```
# 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 change your customer data just add another column in your database table or change an existing column:

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

After the change in the database table you have to adjust the MAP array:

```

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

```

Ofcourse all customer data can be edited or changed via the agent interface.

### 8.2.1.1. Customer with multiple IDs

A customer user can also have more than one customer ID. that might be usefull, if a customer needs to access tickets from another customer user, e.g. all employees of a company want to see the tickets from their colleagues.

Just add another column for the additional ID in your database table like CustomerID1, CustomerID2, CustomerID3", e.g.:

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

After the change in the database table you have to adjust the MAP array and add the new column:

```

# var, frontend, storage, shown (1=always,2=lite), required, storage-type, http

```

```
[ 'UserCustomerIDs', 'CustomerIDs', 'customer_ids', 1, 0, 'var', "", 0 ],
```

Ofcourse the multiple customer IDs can be handled through the webinterface of OTRS.

## 8.2.2. LDAP

If you already have a LDAP directory with customer data you can use it with OTRS ofcourse.

This example shows how to define a LDAP customer user backend in your `Kernel/Config.pm` file.

```
# 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 => '(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=gesperrt))',
# 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 you want to adjust your customer data just add the new values to the MAP array or remove not needed entries.

```

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

```

### 8.2.2.1. Customer with multiple IDs

A customer user can also have more than one customer ID. that might be usefull, if a customer needs to access tickets from another customer user, e.g. all employees of a company want to see the tickets from their colleagues.

Just add another column for the additional ID in your LDAP directory like CustomerID1, CustomerID2, CustomerID3", e.g. After this changes you have to add the new entry to the MAP array:

```

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

```

Ofcourse the multiple customer IDs have to be edited by the LDAP admin, OTRS can only read these data and not write to the LDAP tree.

## 8.3. Customer authentication backend

### 8.3.1. Database (Default)

The default authentication backend for customer users is the OTRS database which can be used through the agent and customer interface. It is possible to create and change the user data via the web interface of OTRS.

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

### 8.3.2. LDAP

If you have a LDAP directory with customer data you can use the LDAP backend to authenticate the customer users to the ticket system. OTRS has only read access to the LDAP tree, that means that no customer user data can be added or changed via the agent or customer interface of the ticket system.

```
# 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,
#   };

```

### 8.3.3. HTTPBasicAuth

If you want to implement a "single sign on" system for your customer users use http basic authentication for all of your systems and activate also this backend for OTRS (no login is for OTRS needed).

```

# 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';

```

### 8.3.4. Radius

The following settings can be used to authenticate against a radius server:

```

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

```

## 8.4. Customize the customer self registration

It is possible to customize the customer self registration for new customer users via `customer.pl`. You can add more optional or necessary fields, e.g. address, city, phone number.

In the following example a new necessary field for the phone number is added.

### 8.4.1. Customize the web interface

Because the new necessary field for the phone number is displayed in the web interface, the `dtl` file for the webinterface must be changed. Edit the file `Kernel/Output/HTML/Standard/CustomerLogin.dtl` and add the new input field for the phone number in line 128.

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

### 8.4.2. Customer mapping

Also a new entry for the phone number has to be added to the customer mapping. Copy "CustomerUser" from `Kernel/Config/Defaults.pm` into the `Kernel/Config.pm` file and add the phone field to the MAP array.

```
# 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"}', 0 ],
    [ '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.',
    },
},
};

```

### 8.4.3. Altering the customer table

Also a new column has to be added to the "customer\_user" table in the database.

```
ALTER TABLE customer_user ADD phone VARCHAR (200);
```

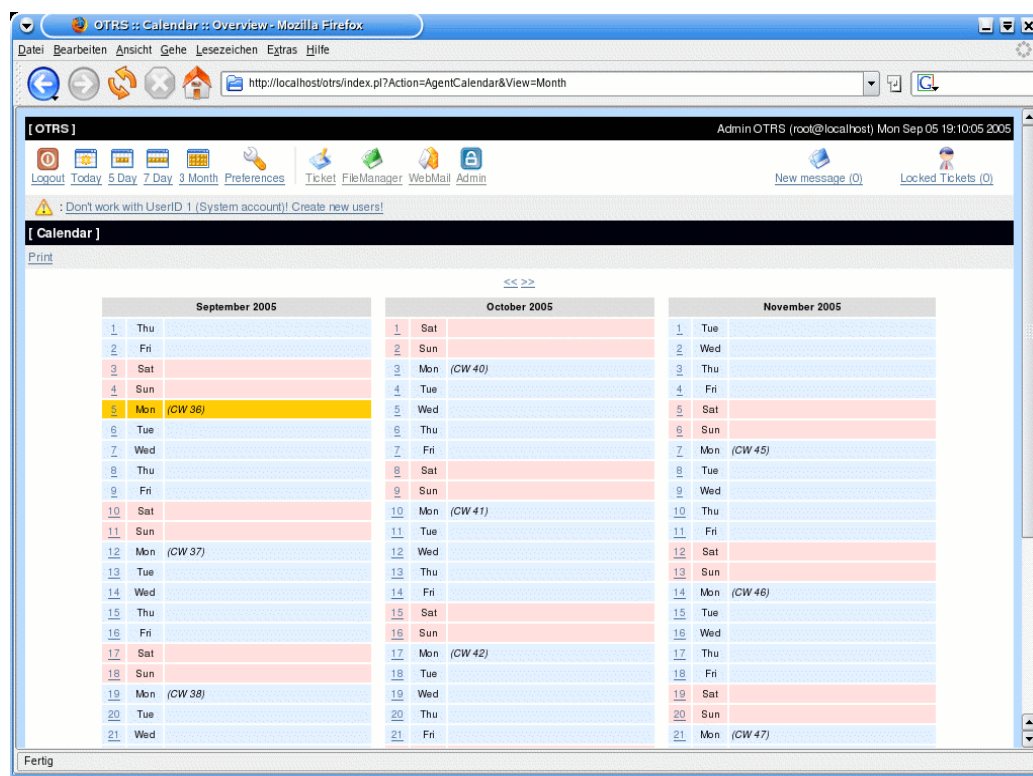
Now all adjustments are made and the new phone number field can be used. Restart your web server if you use mod\_perl and open the customer interface (customer.pl) to check if the new settings are working properly.

# Chapter 9. Additional applications

Since OTRS 2.0 it is possible to install additional applications that can be used with the central framework. The installation can be done with the package manager via the adminarea. The package manager downloads the applications from a online repository and takes care of package dependencies.

## 9.1. Calendar

With the webbased calendar you can edit and view userbased appointments that can be private or public.

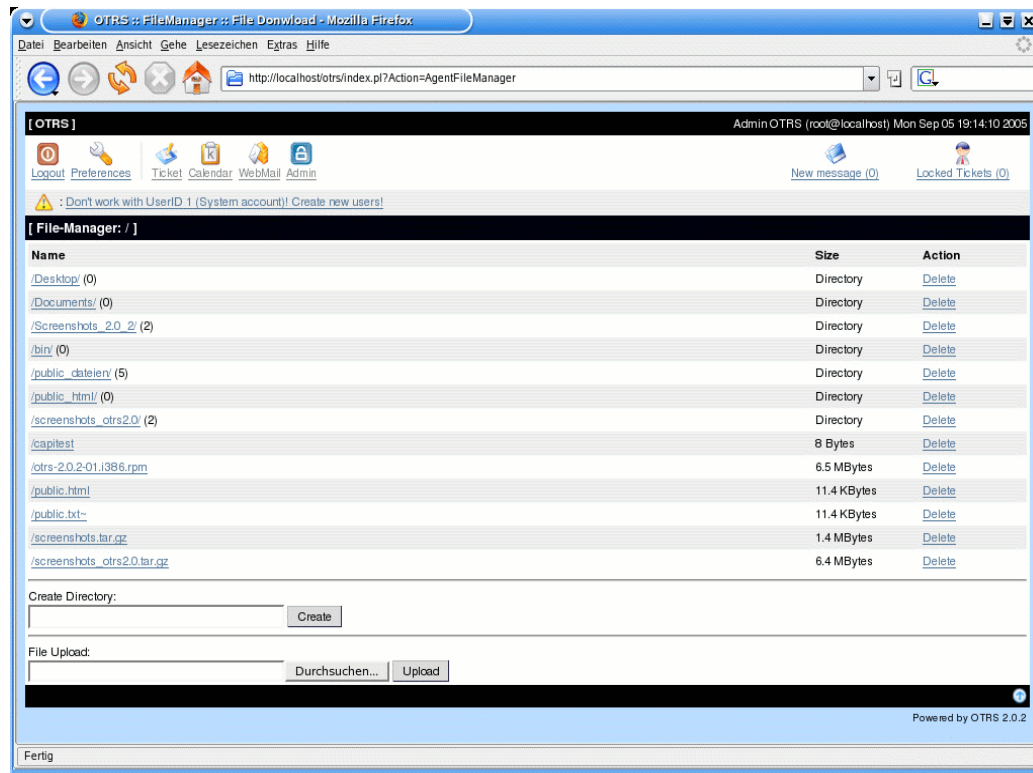


## 9.2. Content manager

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

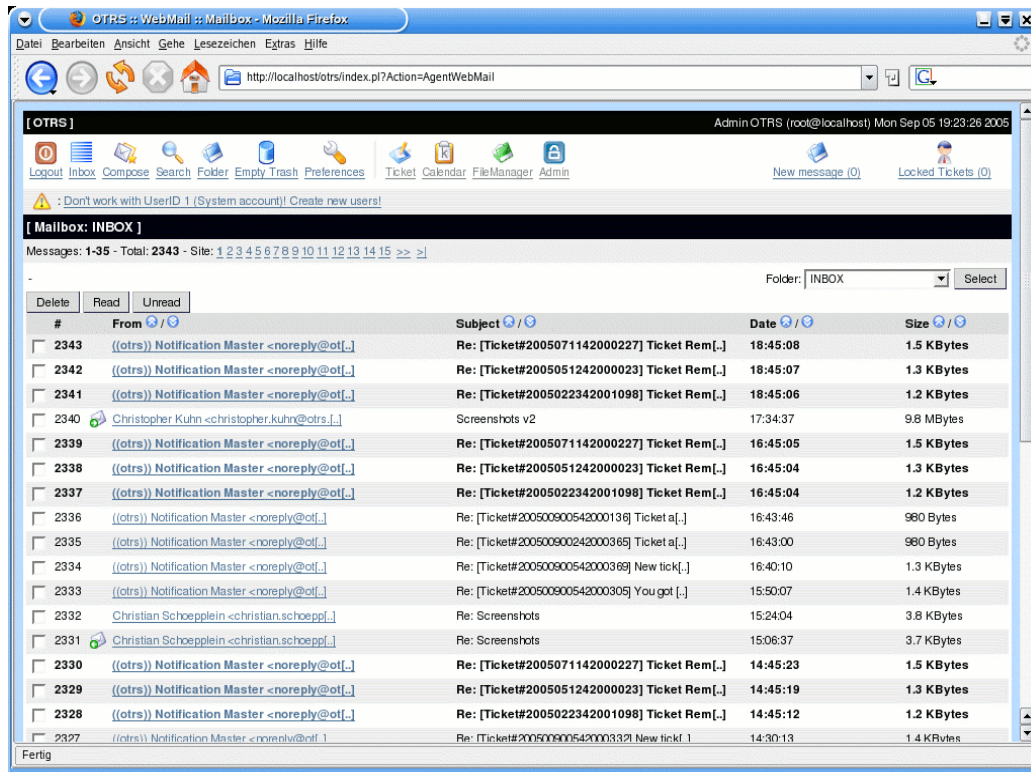
## 9.3. File manager

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, e.g.



## 9.4. 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, e.g., simply with your web browser.



## 9.5. FAQ

The FAQ system is no separate application at present and 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.

## 9.6. System status

With the system status application the OTRS admin can get an overview on 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) ]**

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 10. 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.

## 10.1. OTRS

There are several options to improve the performance via OTRS.

### 10.1.1. 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 having about 60.000 tickets (till 6.000 open tickets) in your system.
- Kernel::System::Ticket::IndexAccelerator::StaticDB, the most powerful module, it should be used 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.

### 10.1.2. TicketStorageModule

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

Kernel/Config.pm

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



[ . . . ]

- Kernel::System::Ticket::ArticleStorageDB (default), store attachments and co in the database. Note: Don't use it for larger setups.

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

Contra: It's not really nice to store attachments in your database. Take care that your database is able to store large objects. E.g. MySQL (config) "set-variable = max\_allowed\_packet=8M" to store 8 MB objects (default is 2M).

- Kernel::System::Ticket::ArticleStorageFS, store attachments and co in local file system. Note: Use it for larger setups.

Pro: Faster!

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

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

## 10.2. 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.

### 10.2.1. MySQL

If you use MySQL table type MyISAM (which is 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;
```

## 10.2.2. 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)

([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>.

## 10.3. 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).

### 10.3.1. Preestablish database connections

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

### 10.3.2. Preloaded modules - startup.pl

Use the scripts/apache-perl-startup.pl (mod\_perl 1.0) bzw. 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).

### 10.3.3. 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 cant use the OTRS-Paket-Manager via web interface, you need to use CMD - bin/opm.pl.

### 10.3.4. Choosing the Right Strategy

If you have a really large installation (over 1000 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)).

### 10.3.5. mod\_gzip/mod\_deflate

Anyway, 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/) (http://www.schroepl.net/projekte/mod\_gzip/)) or mod\_deflate for Apache2 (default module in Apache2). If you have a html page with 45k, mod\_gzip/mod\_deflate compresses it. The page will be about 7k. That's really nice.

### 10.3.6. mod\_dosevasive

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

# Chapter 11. Backing up the system

This chapter describes the backup and restore of the data, that are relevant for OTRS.

## 11.1. 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 were stored into the directory `/backup/2005-09-12_14-28/`. Every type of data was saved into a extra `.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#
```

## 11.2. 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` is shipped with every OTRS installation, to simplify the restore process. 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 last 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.	<a href="http://lists.otrs.org/cgi-bin/listinfo/announce">http://lists.otrs.org/cgi-bin/listinfo/announce</a> ( <a href="http://lists.otrs.org/cgi-bin/listinfo/announce">http://lists.otrs.org/cgi-bin/listinfo/announce</a> )

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

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 know 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, the 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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'HttpType'} = 'http';</code>

**B.1.1.6. ScriptAlias**

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

**B.1.1.7. AdminEmail**

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

Description	Value
Config-Setting:	<code>\$Self-&gt;{'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-&gt;{'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-&gt;{'DefaultCharset'} = 'iso-8859-1';</code>

### B.1.1.10. DefaultLanguage

Description	Value
Description:	The default frontend language. Possible values are bg, cz, de, en, 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-&gt;{'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-&gt;{'DefaultUsedLanguages'} = { 'pt' =&gt; 'Portugu&amp;ecirc;s', 'es' =&gt; 'Espa&amp;ntilde;ol',          'ru' =&gt; 'Russian', 'fr' =&gt; 'Fran&amp;ccedil;ais',      'bg' =&gt; 'Bulgarian', 'sv' =&gt; 'Svenska',              'it' =&gt; 'Italiano', 'hu' =&gt; 'Hungarian', 'zh_CN' =&gt; 'Traditional Chinese', 'nb_NO' =&gt; 'Norsk bokm&amp;aring;l', 'pl' =&gt; 'Polski',              'cz' =&gt; 'Czech', 'nl' =&gt; 'Nederlands',          'fi' =&gt; 'Suomi', 'en' =&gt; 'English',             'de' =&gt; 'Deutsch', 'pt_BR' =&gt; 'Portugu&amp;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-&gt;{'DefaultTheme'} = 'Standard';</pre>

**B.1.1.13. DefaultTheme::HostBased**

Description	Value
Description:	Used default frontend HTML theme based on used web server host name (regexp possible in key).
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self-&gt;{'DefaultTheme::HostBased'} = { 'host2\\example\\.com' =&gt; 'SomeTheme2', 'host1\\example\\.com' =&gt; '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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'NotificationSenderName'} = 'OTRS Notification Master';</code>

#### B.1.1.22. NotificationSenderEmail

Description	Value
Description:	Mail address of sender for notification messages.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'NotificationSenderEmail'} = 'otrs@&lt;OTRS_CONFIG_FQDN&gt;';</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-&gt;{'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-&gt;{'System::Customer::Permission'} = [ 'ro', 'rw' ];</code>

#### B.1.1.25. WebMail::IMAPProtocol

Description	Value
Description:	Default imap protocoll.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'WebMail::IMAPProtocol'} = 'imap';</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



Description	Value
Required:	1
Config-Setting:	<code>\$Self-&gt;{'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
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'MIME-Viewer'}-&gt;{'application/excel'} = 'xlhtml';</code>

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

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

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-&gt;{'MIME-Viewer'}-&gt;{'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-&gt;{'MIME-Viewer'}-&gt;{'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-&gt;{'MIME-Viewer'}-&gt;{'text/xml'} = '&lt;OTRS_CONFIG_Home&gt;/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:	<pre>\$Self-&gt;{'Package::RepositoryRoot'} = [ 'http://ftp.otrs.org/pub/otrs/misc/packages/repository.xml', 'http://otrs.org/repository.xml'      ];</pre>

#### B.1.4.2. Package::RepositoryList

Description	Value
Description:	List of online repositories.
Group:	Framework
SubGroup:	Core::Package
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self-&gt;{'Package::RepositoryList'} = { 'ftp://ftp.example.com/pub/otrs/misc/packages/' =&gt; '[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-&gt;{'Package::Timeout'} = '12';</pre>

#### B.1.4.4. Package::Proxy

Description	Value
Description:	Fetch packages via proxy.
Group:	Framework

Description	Value
SubGroup:	Core::Package
Valid:	0
Required:	0
Config-Setting:	<code>\$Self-&gt;{'Package::Proxy'} = 'http://proxy.sn.no:8001/' ;</code>

## 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-&gt;{'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-&gt;{'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-&gt;{'SendmailModule::Host'} = 'mail.example.com';</code>

#### B.1.5.4. SendmailModule::Port

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-&gt;{'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-&gt;{'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.

Description	Value
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	<code>\$Self-&gt;{'SendmailModule::AuthPassword'} = 'MailserverPassword';</code>

### B.1.5.7. SendmailBcc

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:	0
Required:	0
Config-Setting:	<code>\$Self-&gt;{'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-&gt;{'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.

Description	Value
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'SessionModule'} = 'Kernel::System::AuthSession::DB';</code>

#### B.1.6.2. SessionName

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

#### B.1.6.4. SessionDeleteIfNotRemoteID

Description	Value
Description:	Delete a session, if the session id is used with an invalied remote IP?
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1



Description	Value
Config-Setting:	<code>\$Self-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'SessionDeleteIfTimeToOld'} = '1';</code>

#### B.1.6.8. SessionUseCookie

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

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-&gt;{'SessionUseCookie'} = '1';</code>

#### B.1.6.9. SessionUseCookieAfterBrowserClose

Description	Value
Description:	Store cookies after the browser has ben closed?
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'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-&gt;{'SessionDir'} = '&lt;OTRS_CONFIG_Home&gt;/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

Description	Value
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'SpellChecker'} = '1';</code>

### B.1.7.2. SpellCheckerBin

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-&gt;{'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-&gt;{'SpellCheckerDictDefault'} = 'english';</code>

### B.1.7.4. SpellCheckerIgnore

Description	Value
Description:	A list of words, that are ignored by the spell checker by default.

Description	Value
Group:	Framework
SubGroup:	Core::SpellChecker
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'SpellCheckerIgnore'} = [ 'www', 'webmail', 'https', 'http', 'html', 'rfc' ];</code>

## B.1.8. Core::Time

### B.1.8.1. TimeInputFormat

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-&gt;{'TimeInputFormat'} = 'Option';</code>

### B.1.8.2. TimeCalendarLookup

Description	Value
Description:	Enable a calendar lookup window on date selections.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'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

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'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-&gt;{'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-&gt;{'TimeVacationDays'} = {     '1' =&gt; {         '1' =&gt; 'New Year\'s Eve!'     },     '12' =&gt; {         '25' =&gt; 'First Christmas Day',         '24' =&gt; 'Christmas',         '26' =&gt; 'Second Christmas Day',         '31' =&gt; 'Silvester'     },     '5' =&gt; {         '1' =&gt; '1 St. May'     } }; </pre>

#### B.1.8.6. TimeVacationDaysOneTime

Description	Value
Description:	One time vacation days.
Group:	Framework

Description	Value
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self-&gt;{'TimeVacationDaysOneTime'} = { '2004' =&gt; { '1' =&gt; { '1' =&gt; '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-&gt;{'TimeWorkingHours'} = {     'Mon' =&gt; [         '8',         '9',         '10',         '11',         '12',         '13',         '14',         '15',         '16',         '17',         '18',         '19',         '20'     ],     'Tue' =&gt; [         '8',         '9',         '10',         '11',         '12',         '13',         '14',         '15',         '16',         '17',         '18',         '19',         '20'     ],     'Fri' =&gt; [         '8',         '9',         '10',         '11',         '12',         '13',         '14',         '15',         '16',         '17',         '18',         '19',         '20'     ],     'Wed' =&gt; [         '8',         '9',         '10',         '11',         '12',         '13',         '14',         '15',         '16',         '17',         '18',         '19',         '20'     ],     'Sun' =&gt; [],     'Sat' =&gt; [],     'Thu' =&gt; [         '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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'AttachmentDownloadType'} = 'inline';</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-&gt;{'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-&gt;{'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-&gt;{'Frontend::Output::PostFilter'}-&gt;{'ActiveElementFilter'} = { 'Debug' =&gt; '0', 'Module' =&gt; '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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'PGP::Key::Password'} = { '488A0B8F' =&gt; 'SomePassword', 'D2DF79FA' =&gt; '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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'SMIME::PrivatePath'} = '/etc/ssl/private/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-&gt;{'Frontend::Module'}-&gt;{'Admin'} = { 'NavBar' =&gt; [ 'Prio' =&gt; '10000', 'Block' =&gt; 'ItemArea', 'Image' =&gt; 'admin.png', 'NavBar' =&gt; 'Admin', 'Type' =&gt; 'Menu', 'AccessKey' =&gt; 'a', 'Description' =&gt; 'Admin-Area', 'Name' =&gt; 'Admin', 'Link' =&gt; 'Action=Admin' ], 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin-Area', 'Title' =&gt; 'Admin', 'NavBarModule' =&gt; { 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin' }, 'Group' =&gt; [ ] }; </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-&gt;{'Frontend::Module'}-&gt;{'AdminUser'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin', 'Title' =&gt; 'User', 'NavBarModule' =&gt; { 'Prio' =&gt; '100', 'Block' =&gt; 'Block1', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'Users' }, 'Group' =&gt; [ ] }; </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-&gt;{'Frontend::Module'}-&gt;{'AdminGroup'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin',          'Title' =&gt; 'Group', 'NavBarModule' =&gt; { 'Prio' =&gt; '150', 'Block' =&gt; 'Block1', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'Groups' }, 'Group' =&gt; [ '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-&gt;{'Frontend::Module'}-&gt;{'AdminUserGroup'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin', 'Title' =&gt; 'Users &lt;-&gt; Groups', 'NavBarModule' =&gt; { 'Prio' =&gt; '200', 'Block' =&gt; 'Block1', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'Users &lt;-&gt; Groups' }, 'Group' =&gt; [ '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-&gt;{'Frontend::Module'}-&gt;{'AdminCustomerUser'} = { 'NavBar' =&gt; [ 'Prio' =&gt; '9000', 'Block' =&gt; "", 'Image' =&gt; 'folder_yellow.png', 'NavBar' =&gt; 'Ticket', 'Type' =&gt; "", 'AccessKey' =&gt; 'c', 'Description' =&gt; 'Edit Customer Users', 'Name' =&gt; 'Customer', 'Link' =&gt; 'Action=AdminCustomerUser&amp;Nav=Agent' ], 'GroupRo' =&gt; [ 'NavBarName' =&gt; "", 'Description' =&gt; 'Edit Customer Users', 'Title' =&gt; 'Customer User', 'NavBarModule' =&gt; { 'Prio' =&gt; '300', 'Block' =&gt; 'Block1', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'Customer Users' }, 'Group' =&gt; [ '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-&gt;{'Frontend::Module'}-&gt;{'AdminCustomerUserGroup'} = {   'NavBarName' =&gt; 'Admin',   'Description' =&gt; 'Admin',   'Title' =&gt; 'Customer Users &lt;-&gt; Groups',   'NavBarModule' =&gt; {     'Prio' =&gt; '400',     'Block' =&gt; 'Block1',     'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin',     'Name' =&gt; 'Customer Users &lt;-&gt; Groups'   },   'Group' =&gt; [     '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-&gt;{'Frontend::Module'}-&gt;{'AdminRole'} = {   'NavBarName' =&gt; 'Admin',   'Description' =&gt; 'Admin',   'Title' =&gt; 'Role',   'NavBarModule' =&gt; {     'Prio' =&gt; '500',     'Block' =&gt; 'Block1',     'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin',     'Name' =&gt; 'Roles'   },   'Group' =&gt; [     '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-&gt;{'Frontend::Module'}-&gt;{'AdminRoleUser'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin', 'Title' =&gt; 'Roles &lt;-&gt; Users', 'NavBarModule' =&gt; { 'Prio' =&gt; '600', 'Block' =&gt; 'Block1', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'Roles &lt;-&gt; Users' }, 'Group' =&gt; [ ] }; </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-&gt;{'Frontend::Module'}-&gt;{'AdminRoleGroup'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin', 'Title' =&gt; 'Roles &lt;-&gt; Groups', 'NavBarModule' =&gt; { 'Prio' =&gt; '700', 'Block' =&gt; 'Block1', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'Roles &lt;-&gt; Groups' }, 'Group' =&gt; [ ] }; </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-&gt;{'Frontend::Module'}-&gt;{'AdminSMIME'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin', 'Title' =&gt; 'SMIME Management', 'NavBarModule' =&gt; { 'Prio' =&gt; '500', 'Block' =&gt; 'Block3', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'SMIME' }, 'Group' =&gt; [ '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-&gt;{'Frontend::Module'}-&gt;{'AdminPGP'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin', 'Title' =&gt; 'PGP Key Management', 'NavBarModule' =&gt; { 'Prio' =&gt; '600', 'Block' =&gt; 'Block3', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'PGP' }, 'Group' =&gt; [ '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-&gt;{'Frontend::Module'}-&gt;{'AdminPOP3'} = {   'NavBarName' =&gt; 'Admin',   'Description' =&gt; 'Admin',   'Title' =&gt; 'POP3 Account',          'NavBarModule' =&gt; {   'Prio' =&gt; '100',   'Block' =&gt; 'Block4',   'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin',   'Name' =&gt; 'PostMaster POP3 Account'   },          'Group' =&gt; [   ]          }; </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-&gt;{'Frontend::Module'}-&gt;{'AdminPostMasterFilter'} = {   'NavBarName' =&gt; 'Admin',   'Description' =&gt; 'Admin',   'Title' =&gt; 'PostMaster Filter',   'NavBarModule' =&gt; {   'Prio' =&gt; '200',   'Block' =&gt; 'Block4',   'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin',   'Name' =&gt; 'PostMaster Filter'   },          'Group' =&gt; [   ]          }; </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-&gt;{'Frontend::Module'}-&gt;{'AdminEmail'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin', 'Title' =&gt; 'Admin-Email',           'NavBarModule' =&gt; { 'Prio' =&gt; '400', 'Block' =&gt; 'Block4', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'Admin Notification' },           'Group' =&gt; [ ]           }; </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-&gt;{'Frontend::Module'}-&gt;{'AdminSession'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin', 'Title' =&gt; 'Session Management', 'NavBarModule' =&gt; { 'Prio' =&gt; '500', 'Block' =&gt; 'Block4', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'Session Management' },           'Group' =&gt; [ ]           }; </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-&gt;{'Frontend::Module'}-&gt;{'AdminLog'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin', 'Title' =&gt; 'System Log',           'NavBarModule' =&gt; { 'Prio' =&gt; '600', 'Block' =&gt; 'Block4', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'System Log'           }, 'Group' =&gt; [                     '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-&gt;{'Frontend::Module'}-&gt;{'AdminSelectBox'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin', 'Title' =&gt; 'Select box',           'NavBarModule' =&gt; { 'Prio' =&gt; '700', 'Block' =&gt; 'Block4', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'Select Box'           }, 'Group' =&gt; [                     '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-&gt;{'Frontend::Module'}-&gt;{'AdminPackageManager'} = {   'NavBarName' =&gt; 'Admin',   'Description' =&gt; 'Software Package Manager',   'Title' =&gt; 'Package Manager',   'NavBarModule' =&gt; {     'Prio' =&gt; '1000',     'Block' =&gt; 'Block4',     'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin',     'Name' =&gt; 'Package Manager'   },   'Group' =&gt; [     'admin'   ]; </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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'NotificationBodyLostPassword'} = 'Hi &lt;OTRS_USERFIRSTNAME&gt;, you or someone impersonating you has requested to change your OTRS password. New Password: &lt;OTRS_NEWPW&gt; &lt;OTRS_CONFIG_HttpType&gt;://&lt;OTRS_CONFIG_FQDN&gt;/&lt;OTRS_CONFIG_ScriptAlias&gt;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-&gt;{'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-&gt;{'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-&gt;{'UserSyncLDAPMap'} = {   'Email' =&gt; 'mail',           'Lastname' =&gt; 'sn',   'Firstname' =&gt; '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-&gt;{'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-&gt;{'Frontend::NotifyModule'}-&gt;{'1-CharsetCheck'} = {   'Module' =&gt; '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-&gt;{'Frontend::NotifyModule'}-&gt;{'2-UID-Check'} = { 'Module' =&gt; '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-&gt;{'Frontend::NotifyModule'}-&gt;{'3-ShowAgentOnline'} = { 'Module' =&gt; '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-&gt;{'Frontend::NotifyModule'}-&gt;{'4-ShowCustomerOnline'} = { 'Module' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'Logout'} = {   'NavBar' =&gt; [     {       'Prio' =&gt; '100',       'Block' =&gt; 'ItemPre',       'Image' =&gt; 'exit.png',       'NavBar' =&gt; "",       'Type' =&gt; "",       'AccessKey' =&gt; 'l',       'Description' =&gt; 'Logout',       'Name' =&gt; 'Logout',       'Link' =&gt; 'Action=Logout'     }   ],   'NavBarName' =&gt; "",   'Description' =&gt; 'Logout',   'Title' =&gt; "" }; </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-&gt;{'Frontend::Module'}-&gt;{'AgentPreferences'} = {   'NavBar' =&gt; [     {       'Prio' =&gt; '1000',       'Block' =&gt; "",       'Image' =&gt; 'prefer.png',       'NavBar' =&gt; "",       'Type' =&gt; "",       'AccessKey' =&gt; 'p',       'Description' =&gt; 'Agent Preferences',       'Name' =&gt; 'Preferences',       'Link' =&gt; 'Action=AgentPreferences'     }   ],   'NavBarName' =&gt; "",   'Description' =&gt; 'Agent Preferences',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentSpelling'} = {   'NavBarName' =&gt; "",   'Description' =&gt; 'Spell checker',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentBook'} = {   'NavBarName' =&gt; "",   'Description' =&gt; 'Address book of CustomerUser sources',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentLookup'} = {   'NavBarName' =&gt; "",   'Description' =&gt; 'Data table lookup module.',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentLinkObject'} = {   'NavBarName' =&gt; "",   'Description' =&gt; 'Link Object',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentInfo'} = {   'NavBarName' =&gt; "",   'Description' =&gt; 'Generic Info module',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentCalendarSmall'} = {   'NavBarName' =&gt; "",   'Description' =&gt; 'Small calendar for date selection.',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'SystemStats'} = {   'NavBar' =&gt; [     {       'Prio' =&gt; '400',       'Block' =&gt; "",       'Image' =&gt; 'stats.png',       'NavBar' =&gt; 'Ticket',       'Type' =&gt; "",       'AccessKey' =&gt; 't',       'Description' =&gt; 'Stats-Area',       'Name' =&gt; 'Stats',       'Link' =&gt; 'Action=SystemStats'     }   ],   'GroupRo' =&gt; [     'stats'   ],   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Stats',   'Title' =&gt; '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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'PreferencesGroups'}-&gt;{'Password'} = { 'Prio' =&gt; '1000', 'PasswordNeedDigit' =&gt; '0', 'Label' =&gt; 'Change Password', 'PasswordMin2Characters' =&gt; '0', 'PasswordRegExp' =&gt; "", 'Module' =&gt; 'Kernel::Output::HTML::PreferencesPassword', 'PasswordMin2Lower2UpperCharacters' =&gt; '0', 'Area' =&gt; 'Agent', 'Activ' =&gt; '1', 'Colum' =&gt; 'Other Options', 'PasswordMinSize' =&gt; '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-&gt;{'PreferencesGroups'}-&gt;{'SpellDict'} = { 'Prio' =&gt; '5000', 'Label' =&gt; 'Spelling Dictionary', 'PrefKey' =&gt; 'UserSpellDict', 'Module' =&gt; 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' =&gt; 'Select your default spelling dictionary.', 'Activ' =&gt; '1', 'Data' =&gt; { 'deutsch' =&gt; 'Deutsch', 'english' =&gt; 'English' }, 'Colum' =&gt; '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-&gt;{'PreferencesGroups'}-&gt;{'Comment'} = { 'Prio' =&gt; '6000',           'Block' =&gt; 'Input', 'Label' =&gt; 'Comment', 'PrefKey' =&gt; 'UserComment', 'Module' =&gt; 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' =&gt; 'Comment',       'Activ' =&gt; '0', 'Data' =&gt; '\$Env{"UserComment"}', 'Colum' =&gt; '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-&gt;{'PreferencesGroups'}-&gt;{'FreeText'} = { 'Prio' =&gt; '7000',           'Block' =&gt; 'Input', 'Label' =&gt; 'Comment', 'PrefKey' =&gt; 'UserFreeText', 'Module' =&gt; 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' =&gt; 'Example for free text', 'Activ' =&gt; '1', 'Data' =&gt; '\$Env{"UserFreeText"}', 'Colum' =&gt; '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-&gt;{'PreferencesGroups'}-&gt;{'Language'} = { 'Prio' =&gt; '1000',          'Label' =&gt; 'Language', 'PrefKey' =&gt; 'UserLanguage', 'Module' =&gt; 'Kernel::Output::HTML::PreferencesLanguage', 'Desc' =&gt; 'Select your frontend language.', 'Activ' =&gt; '1',           'Colum' =&gt; 'Frontend' };</pre>

### B.1.17.9. PreferencesGroups###Theme

Description	Value
Description:	Parameters for the Theme object in the preference view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self-&gt;{'PreferencesGroups'}-&gt;{'Theme'} = { 'Prio' =&gt; '2000',          'Label' =&gt; 'Theme', 'PrefKey' =&gt; 'UserTheme', 'Module' =&gt; 'Kernel::Output::HTML::PreferencesTheme', 'Desc' =&gt; 'Select your frontend Theme.', 'Activ' =&gt; '1',           'Colum' =&gt; '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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'CustomerPanelBodyLostPassword'} = 'Hi &lt;OTRS_USERFIRSTNAME&gt;, you or someone impersonating you has requested to change your OTRS password. New Password: &lt;OTRS_NEWPW&gt; &lt;OTRS_CONFIG_HttpType&gt;://&lt;OTRS_CONFIG_FQDN&gt;/&lt;OTRS_CONFIG_ScriptAlias&gt;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-&gt;{'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-&gt;{'CustomerPanelBodyNewAccount'} = 'Hi &lt;OTRS_USERFIRSTNAME&gt;, you or someone impersonating you has created a new OTRS account for you (&lt;OTRS_USERFIRSTNAME&gt; &lt;OTRS_USERLASTNAME&gt;). Login: &lt;OTRS_USERLOGIN&gt; Password: &lt;OTRS_USERPASSWORD&gt; &lt;OTRS_CONFIG_HttpType&gt;://&lt;OTRS_CONFIG_FQDN&gt;/&lt;OTRS_CONFIG_ScriptAlias&gt;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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'Customer::AuthModule::LDAP::AlwaysFilter'} = '(!objectclass=c</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-&gt;{'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-&gt;{'Customer::AuthModule::LDAP::Params'} = {   'async' =&gt; '0',          'timeout' =&gt; '120',   'version' =&gt; '3',        'port' =&gt; '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-&gt;{'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-&gt;{'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-&gt;{'CustomerFrontend::NotifyModule'}-&gt;{'1-ShowAgentOnline'} = {   'Module' =&gt; '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-&gt;{'CustomerFrontend::NotifyModule'}-&gt;{'1-ShowCustomerOnline'} =   'Module' =&gt; '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-&gt;{'CustomerFrontend::Module'}-&gt;{'Logout'} = { 'NavBar' =&gt; [ 'Prio' =&gt; '10', 'Image' =&gt; 'exit.png', 'NavBar' =&gt; "", 'AccessKey' =&gt; 'l', 'Description' =&gt; 'Logout', 'Name' =&gt; 'Logout', 'Link' =&gt; 'Action=Logout' ], 'NavBarName' =&gt; "", 'Description' =&gt; 'Logout of customer panel', 'Title' =&gt; "" }; </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-&gt;{'CustomerFrontend::Module'}-&gt;{'CustomerPreferences'} = { 'NavBar' =&gt; [ 'Prio' =&gt; '1000', 'Image' =&gt; 'prefer.png', 'NavBar' =&gt; "", 'AccessKey' =&gt; 'p', 'Description' =&gt; 'Preferences', 'Name' =&gt; 'Preferences', 'Link' =&gt; 'Action=CustomerPreferences' ] }, 'NavBarName' =&gt; "", 'Description' =&gt; 'Customer preferences', 'Title' =&gt; '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-&gt;{'CustomerFrontend::Module'}-&gt;{'CustomerCalendarSmall'} = {   'NavBarName' =&gt; "",   'Description' =&gt; 'Small calendar for date selection.',   'Title' =&gt; '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-&gt;{'CustomerFrontend::Module'}-&gt;{'CustomerAccept'} = {   'NavBarName' =&gt; "",   'Description' =&gt; 'To accept login infos',   'Title' =&gt; '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-&gt;{'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-&gt;{'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-&gt;{'CustomerPreferences'} = {   'Params' =&gt; {     'TableUserID' =&gt; 'user_id',     'TableValue' =&gt; 'preferences_value',     'TableKey' =&gt; 'preferences_key',     'Table' =&gt; 'customer_preferences'   },   'Module' =&gt; '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-&gt;{'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-&gt;{'CustomerPreferencesGroups'}-&gt;{'Password'} = { 'Prio' =&gt; '1000',          'PasswordNeedDigit' =&gt; '0', 'Label' =&gt; 'Change Password', 'PasswordMin2Characters' =&gt; '0', 'PasswordRegExp' =&gt; "", 'Module' =&gt; 'Kernel::Output::HTML::PreferencesPassword', 'PasswordMin2Lower2UpperCharacters' =&gt; '0', 'Area' =&gt; 'Customer',      'Activ' =&gt; '1', 'Colum' =&gt; 'Other Options', 'PasswordMinSize' =&gt; '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-&gt;{'CustomerPreferencesGroups'}-&gt;{'Language'} = { 'Prio' =&gt; '2000',          'Label' =&gt; 'Language', 'PrefKey' =&gt; 'UserLanguage', 'Module' =&gt; 'Kernel::Output::HTML::PreferencesLanguage', 'Desc' =&gt; 'Select your frontend language.', 'Activ' =&gt; '1',           'Colum' =&gt; '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-&gt;{'CustomerPreferencesGroups'}-&gt;{'Theme'} = { 'Prio' =&gt; '1000',           'Label' =&gt; 'Theme', 'PrefKey' =&gt; 'UserTheme', 'Module' =&gt; 'Kernel::Output::HTML::PreferencesTheme', 'Desc' =&gt; 'Select your frontend Theme.', 'Activ' =&gt; '0',           'Colum' =&gt; '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-&gt;{'CustomerPreferencesGroups'}-&gt;{'PGP'} = { 'Prio' =&gt; '10000',           'Label' =&gt; 'PGP Key', 'PrefKey' =&gt; 'UserPGPKey', 'Module' =&gt; 'Kernel::Output::HTML::PreferencesPGP', 'Desc' =&gt; 'PGP Key Upload',           'Activ' =&gt; '1', 'Colum' =&gt; '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-&gt;{'CustomerPreferencesGroups'}-&gt;{'SMIME'} = { 'Prio' =&gt; '11000', 'Label' =&gt; 'SMIME Certificate', 'PrefKey' =&gt; 'UserSMIMEKey', 'Module' =&gt; 'Kernel::Output::HTML::PreferencesSMIME', 'Desc' =&gt; 'SMIME Certificate Upload', 'Activ' =&gt; '1', 'Colum' =&gt; '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-&gt;{'LinkObject'}-&gt;{'Ticket'} = { 'Type' =&gt; 'Object', 'LinkObjects' =&gt; [ 'Ticket', 'FAQ' ], 'Name' =&gt; '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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'LoopProtectionLog'} = '&lt;OTRS_CONFIG_Home&gt;/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-&gt;{'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-&gt;{'PostmasterFollowUpSearchInReferences'} = '0';</code>

**B.2.2.8. PostmasterUserID**

Description	Value
Description:	Specify the postmaster DB userID.



Description	Value
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-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-&gt;{'PostMaster::PreFilterModule'}-&gt;{'1-Match'} = {   'Match' =&gt; {     'From' =&gt; 'noreply@'   },   'Set' =&gt; {     'X-OTRS-Ignore' =&gt; 'yes'   },   'Module' =&gt; '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-&gt;{'PostMaster::PreFilterModule'}-&gt;{'2-Match'} = {   'Match' =&gt; {     'Subject' =&gt; 'SomeNumber:(\\d\\d\\d\\d)'   },   'Set' =&gt; {     'X-OTRS-TicketKey-1' =&gt; 'SomeNumber',     'X-OTRS-TicketValue-1' =&gt; '[***]'   },   'Module' =&gt; '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-&gt;{'PostMaster::PreFilterModule'}-&gt;{'5-SpamAssassin'} = {   'Set' =&gt; {     'X-OTRS-Queue' =&gt; 'spam'   },   'CMD' =&gt; '/usr/bin/spamassassin   grep -i "X-Spam-Status: yes"',   'Module' =&gt; '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-&gt;{'PostMaster::PreFilterModule'}-&gt;{'000-MatchDBSource'} = {   'Module' =&gt; '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-&gt;{'PostMaster::PreFilterModule'}-&gt;{'999-AgentInterface'} = {   'Module' =&gt; 'Kernel::System::PostMaster::Filter::AgentInterface',   'AgentInterfaceAddress' =&gt; '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-&gt;{'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-&gt;{'SystemStatsMap'}-&gt;{'Ticket::Stats1'} = {  'UseResultCache' =&gt; '0', 'Output' =&gt; [  'Print', 'CSV',  'Graph' ],  'Module' =&gt; 'Kernel::System::Stats::NewTickets',  'Desc' =&gt; 'New created tickets for each queue in selected month',  'SumRow' =&gt; '1', 'SumCol' =&gt; '1',  'OutputDefault' =&gt; 'Print',  'Name' =&gt; '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-&gt;{'SystemStatsMap'}-&gt;{'Ticket::Stats2'} = {   'UseResultCache' =&gt; '1',   'Output' =&gt; [     'Print',     'CSV',     'Graph'   ],   'Module' =&gt; 'Kernel::System::Stats::TicketOverview',   'Desc' =&gt; 'Overview of the tickets in queue at the end of this month',   'SumRow' =&gt; '1',   'SumCol' =&gt; '1',   'OutputDefault' =&gt; 'Print',   'Name' =&gt; '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-&gt;{'SystemStatsMap'}-&gt;{'Ticket::Stats4'} = {   'UseResultCache' =&gt; '0',   'Output' =&gt; [     'Print',     'CSV',     'Graph'   ],   'Module' =&gt; 'Kernel::System::Stats::StateAction',   'Desc' =&gt; 'Trace system activities (Replacement of old bin/mkStats.pl',   'SumRow' =&gt; '1',   'SumCol' =&gt; '1',   'OutputDefault' =&gt; 'Graph',   'Name' =&gt; '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-&gt;{'SystemStatsMap'}-&gt;{'Ticket::Stats5'} = {   'UseResultCache' =&gt; '0',   'Output' =&gt; [     'Print',     'CSV'   ],   'Module' =&gt; 'Kernel::System::Stats::AccountedTime',   'Desc' =&gt; 'A list about accounted time per customer',   'OutputDefault' =&gt; 'Print',   'Name' =&gt; '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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'Ticket::ForceNewStateAfterLock'} = { 'new' =&gt; '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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'Ticket::CounterLog'} = '&lt;OTRS_CONFIG_Home&gt;/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-&gt;{'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-&gt;{'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-&gt;{'ArticleDir'} = '&lt;OTRS_CONFIG_Home&gt;/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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'Ticket::StateAfterPending'} = { 'pending auto close-' =&gt; 'closed unsuccessful', 'pending auto close+' =&gt; '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-&gt;{'System::Permission'} = [     'ro',     '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-&gt;{'Ticket::Permission'}-&gt;{'1-OwnerCheck'} = {     'Granted' =&gt; '0',     'Module' =&gt; 'Kernel::System::Ticket::Permission::OwnerCheck',     'Required' =&gt; '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-&gt;{'Ticket::Permission'}-&gt;{'2-GroupCheck'} = {     'Granted' =&gt; '0',     'Module' =&gt; 'Kernel::System::Ticket::Permission::GroupCheck',     'Required' =&gt; '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-&gt;{'CustomerTicket::Permission'}-&gt;{'1-CustomerUserIDCheck'} = {   'Granted' =&gt; '1',   'Module' =&gt; 'Kernel::System::Ticket::CustomerPermission::CustomerUserIDCheck',   'Required' =&gt; '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-&gt;{'CustomerTicket::Permission'}-&gt;{'2-CustomerIDCheck'} = {   'Granted' =&gt; '0',   'Module' =&gt; 'Kernel::System::Ticket::CustomerPermission::CustomerIDCheck',   'Required' =&gt; '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-&gt;{'CustomerTicket::Permission'}-&gt;{'3-GroupCheck'} = {   'Granted' =&gt; '0',   'Module' =&gt; 'Kernel::System::Ticket::CustomerPermission::GroupCheck',   'Required' =&gt; '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-&gt;{'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-&gt;{'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-&gt;{'TicketFreeKey1'} = { 'Product' =&gt; '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-&gt;{'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-&gt;{'TicketFreeText1'} = {  'Notebook' =&gt; 'Notebook',  'PC' =&gt; 'PC',  'Phone' =&gt; '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-&gt;{'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' };i

### 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-&gt;{ '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-&gt;{'TicketFreeText2'} = {     'Notebook' =&gt; 'Notebook',     'PC' =&gt; 'PC',     'Phone' =&gt; '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-&gt;{'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-&gt;{'TicketFreeKey3'} = {     'Product' =&gt; '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-&gt;{'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-&gt;{'TicketFreeText3'} = {     'Notebook' =&gt; 'Notebook',     'PC' =&gt; 'PC',     'Phone' =&gt; '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-&gt;{'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-&gt;{'TicketFreeKey4'} = {     'Product' =&gt; '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-&gt;{'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-&gt;{'TicketFreeText4'} = {     'Notebook' =&gt; 'Notebook',     'PC' =&gt; 'PC',     'Phone' =&gt; '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-&gt;{'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-&gt;{'TicketFreeKey5'} = { 'Product' =&gt; '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-&gt;{'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-&gt;{'TicketFreeText5'} = {     'Notebook' =&gt; 'Notebook',     'PC' =&gt; 'PC',     'Phone' =&gt; '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-&gt;{'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-&gt;{'TicketFreeKey6'} = {     'Product' =&gt; '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-&gt;{'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-&gt;{'TicketFreeText6'} = {     'Notebook' =&gt; 'Notebook',     'PC' =&gt; 'PC',     'Phone' =&gt; '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-&gt;{'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-&gt;{'TicketFreeKey7'} = {     'Product' =&gt; '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-&gt;{'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-&gt;{'TicketFreeText7'} = {  'Notebook' =&gt; 'Notebook',  'Phone' =&gt; '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-&gt;{'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-&gt;{'TicketFreeKey8'} = { 'Product' =&gt; '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-&gt;{ '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-&gt;{'TicketFreeText8'} = {     'Notebook' =&gt; 'Notebook',     'PC' =&gt; 'PC',     'Phone' =&gt; '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-&gt;{'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-&gt;{'Frontend::Module'}-&gt;{'AdminQueue'} = {   'NavBarName' =&gt; 'Admin',   'Description' =&gt; 'Admin',           'Title' =&gt; 'Queue',   'NavBarModule' =&gt; {     'Prio' =&gt; '100',     'Block' =&gt; 'Block2',     'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin',     'Name' =&gt; 'Queue'   },   'Group' =&gt; [                        '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-&gt;{'Frontend::Module'}-&gt;{'AdminResponse'} = {   'NavBarName' =&gt; 'Admin',   'Description' =&gt; 'Admin',   'Title' =&gt; 'Response',           'NavBarModule' =&gt; {     'Prio' =&gt; '200',     'Block' =&gt; 'Block2',     'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin',     'Name' =&gt; 'Responses'   },   'Group' =&gt; [                        '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-&gt;{'Frontend::Module'}-&gt;{'AdminQueueResponses'} = {   'NavBarName' =&gt; 'Admin',   'Description' =&gt; 'Admin',   'Title' =&gt; 'Responses &lt;-&gt; Queue',   'NavBarModule' =&gt; {     'Prio' =&gt; '300',     'Block' =&gt; 'Block2',     'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin',     'Name' =&gt; 'Responses &lt;-&gt; Queue'   },   'Group' =&gt; [     '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-&gt;{'Frontend::Module'}-&gt;{'AdminAutoResponse'} = {   'NavBarName' =&gt; 'Admin',   'Description' =&gt; 'Admin',   'Title' =&gt; 'Auto response change',   'NavBarModule' =&gt; {     'Prio' =&gt; '400',     'Block' =&gt; 'Block2',     'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin',     'Name' =&gt; 'Auto Responses'   },   'Group' =&gt; [     '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-&gt;{'Frontend::Module'}-&gt;{'AdminQueueAutoResponse'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin', 'Title' =&gt; 'Auto Responses &lt;-&gt; Queue', 'NavBarModule' =&gt; { 'Prio' =&gt; '500', 'Block' =&gt; 'Block2', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'Auto Responses &lt;-&gt; Queue' }, 'Group' =&gt; [ ] }; </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-&gt;{'Frontend::Module'}-&gt;{'AdminAttachment'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin', 'Title' =&gt; 'Attachment', 'NavBarModule' =&gt; { 'Prio' =&gt; '600', 'Block' =&gt; 'Block2', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'Attachments' }, 'Group' =&gt; [ ] }; </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-&gt;{'Frontend::Module'}-&gt;{'AdminResponseAttachment'} = {   'NavBarName' =&gt; 'Admin',   'Description' =&gt; 'Admin',   'Title' =&gt; 'Attachments &lt;-&gt; Responses',   'NavBarModule' =&gt; {     'Prio' =&gt; '700',     'Block' =&gt; 'Block2',     'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin',     'Name' =&gt; 'Attachments &lt;-&gt; Responses'   },   'Group' =&gt; [     '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-&gt;{'Frontend::Module'}-&gt;{'AdminSalutation'} = {   'NavBarName' =&gt; 'Admin',   'Description' =&gt; 'Admin',   'Title' =&gt; 'Salutation',   'NavBarModule' =&gt; {     'Prio' =&gt; '100',     'Block' =&gt; 'Block3',     'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin',     'Name' =&gt; 'Salutations'   },   'Group' =&gt; [     '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-&gt;{'Frontend::Module'}-&gt;{'AdminSignature'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin', 'Title' =&gt; 'Signature',           'NavBarModule' =&gt; { 'Prio' =&gt; '200', 'Block' =&gt; 'Block3', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'Signatures'           }, 'Group' =&gt; [                     '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-&gt;{'Frontend::Module'}-&gt;{'AdminSystemAddress'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin', 'Title' =&gt; 'System address', 'NavBarModule' =&gt; { 'Prio' =&gt; '300', 'Block' =&gt; 'Block3', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'Email Addresses' },           'Group' =&gt; [ ]                                '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-&gt;{'Frontend::Module'}-&gt;{'AdminNotification'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin', 'Title' =&gt; 'Notification',          'NavBarModule' =&gt; { 'Prio' =&gt; '400', 'Block' =&gt; 'Block3', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'Notifications'          }, 'Group' =&gt; [                      '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-&gt;{'Frontend::Module'}-&gt;{'AdminState'} = { 'NavBarName' =&gt; 'Admin', 'Description' =&gt; 'Admin',          'Title' =&gt; 'State', 'NavBarModule' =&gt; { 'Prio' =&gt; '700', 'Block' =&gt; 'Block3', 'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' =&gt; 'Status'                  }, 'Group' =&gt; [                      '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-&gt;{'Frontend::Module'}-&gt;{'AdminGenericAgent'} = {   'NavBarName' =&gt; 'Admin',   'Description' =&gt; 'Admin',   'Title' =&gt; 'GenericAgent',           'NavBarModule' =&gt; {   'Prio' =&gt; '300',   'Block' =&gt; 'Block4',   'Module' =&gt; 'Kernel::Output::HTML::NavBarModuleAdmin',   'Name' =&gt; 'GenericAgent'           },   'Group' =&gt; [                        '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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'Ticket::Frontend::NewQueueSelectionString'} = '&lt;Queue&gt;';</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-&gt;{'Ticket::Frontend::NewQueueOwnSelection'} = { '2' =&gt; 'Second Queue!', '1' =&gt; '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-&gt;{'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-&gt;{'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-&gt;{'Frontend::CommonObject'}-&gt;{'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-&gt;{'Frontend::CommonObject'}-&gt;{'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-&gt;{'Frontend::CommonParam'}-&gt;{'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-&gt;{'Frontend::CommonParam'}-&gt;{'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-&gt;{'Frontend::CommonParam'}-&gt;{'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-&gt;{'Frontend::NotifyModule'}-&gt;{'3-Ticket::AgentTicketSeen'} = {   'Module' =&gt; '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-&gt;{'Frontend::NotifyModule'}-&gt;{'4-Ticket::TicketNotify'} = {   'Module' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketQueue'} = { 'NavBar' =&gt; [ 'Prio' =&gt; '100', 'Block' =&gt; "", 'Image' =&gt; 'overview.png', 'NavBar' =&gt; 'Ticket', 'Type' =&gt; "", 'AccessKey' =&gt; 'o', 'Description' =&gt; 'Overview of all open Tickets', 'Name' =&gt; 'QueueView', 'Link' =&gt; 'Action=AgentTicketQueue' }, { 'Prio' =&gt; '200', 'Block' =&gt; 'ItemArea', 'Image' =&gt; 'desktop.png', 'NavBar' =&gt; 'Ticket', 'Type' =&gt; 'Menu', 'AccessKey' =&gt; 't', 'Description' =&gt; 'Ticket-Area', 'Name' =&gt; 'Ticket', 'Link' =&gt; 'Action=AgentTicketQueue' } ], 'NavBarName' =&gt; 'Ticket', 'Description' =&gt; 'Overview of all open Tickets', 'Title' =&gt; 'QueueView' }; </pre>

### B.2.9.2. Frontend::Module###AgentTicketPhone

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

Description	Value
Config-Setting:	<pre> \$Self-&gt;{'Frontend::Module'}-&gt;{'AgentTicketPhone'} = { 'NavBar' =&gt; [ 'Prio' =&gt; '200', 'Block' =&gt; "", 'Image' =&gt; 'phone-new.png', 'NavBar' =&gt; 'Ticket', 'Type' =&gt; "", 'AccessKey' =&gt; 'n', 'Description' =&gt; 'Create new Phone Ticket', 'Name' =&gt; 'Phone-Ticket', 'Link' =&gt; 'Action=AgentTicketPhone' ] 'NavBarName' =&gt; 'Ticket', 'Description' =&gt; 'Create new Phone Ticket', 'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketEmail'} = { 'NavBar' =&gt; [ 'Prio' =&gt; '210', 'Block' =&gt; "", 'Image' =&gt; 'mail_new.png', 'NavBar' =&gt; 'Ticket', 'Type' =&gt; "", 'AccessKey' =&gt; 'e', 'Description' =&gt; 'Send Email and create a new Ticket', 'Name' =&gt; 'Email-Ticket', 'Link' =&gt; 'Action=AgentTicketEmail' ] 'NavBarName' =&gt; 'Ticket', 'Description' =&gt; 'Create new Email Ticket', 'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketSearch'} = {   'NavBar' =&gt; [     {       'Prio' =&gt; '300',       'Block' =&gt; "",       'Image' =&gt; 'search.png',       'NavBar' =&gt; 'Ticket',       'Type' =&gt; "",       'AccessKey' =&gt; 's',       'Description' =&gt; 'Search Tickets',       'Name' =&gt; 'Search',       'Link' =&gt; 'Action=AgentTicketSearch'     }   ],   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Search Tickets',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketMailbox'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Agent Mailbox',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketStatusView'} = {   'NavBar' =&gt; [     {       'Prio' =&gt; '110',       'Block' =&gt; "",       'Image' =&gt; 'overview.png',       'NavBar' =&gt; 'Ticket',       'Type' =&gt; "",       'AccessKey' =&gt; 'v',       'Description' =&gt; 'Overview of all open Tickets',       'Name' =&gt; 'StatusView',       'Link' =&gt; 'Action=AgentTicketStatusView'     }   ],   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Overview of all open tickets',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentZoom'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'compat module for Ticket Zoom',   'Title' =&gt; "" }; </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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketZoom'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket Zoom',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketAttachment'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'To download attachments',   'Title' =&gt; " };</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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketPlain'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket plain view of an email',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketNote'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket Note',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketMerge'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket Merge',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketPending'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket Pending',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketPriority'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket Priority',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketLock'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket Lock',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketMove'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket Move',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketHistory'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket History',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketOwner'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket Owner',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketCompose'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket Compose Email Answer',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketCustomerFollowUp'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Used if a agent can also be a customer',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketBounce'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket Compose Bounce Email',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketForward'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket Forward Email',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketCustomer'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket Customer',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketClose'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket Close',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketFreeText'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket FreeText',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketPrint'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket Print',   'Title' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'AgentTicketBulk'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket bulk module',   'Title' =&gt; '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-&gt;{'Frontend::NavBarModule'}-&gt;{'1-Ticket::LockedTickets'} = {   'Module' =&gt; '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-&gt;{'Frontend::NavBarModule'}-&gt;{'2-Ticket::BulkAction'} = {   'Module' =&gt; '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-&gt;{'PreferencesGroups'}-&gt;{'NewTicketNotify'} = { 'Prio' =&gt; '1000', 'Label' =&gt; 'New ticket notification', 'PrefKey' =&gt; 'UserSendNewTicketNotification', 'Module' =&gt; 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' =&gt; 'Send me a notification if there is a new ticket in "My Queue"', 'Activ' =&gt; '1', 'Data' =&gt; { '1' =&gt; 'Yes', '0' =&gt; 'No' }, 'Colum' =&gt; '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-&gt;{'PreferencesGroups'}-&gt;{'FollowUpNotify'} = { 'Prio' =&gt; '2000', 'Label' =&gt; 'Follow up notification', 'PrefKey' =&gt; 'UserSendFollowUpNotification', 'Module' =&gt; 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' =&gt; 'Send me a notification if a customer sends a follow up and I', 'Activ' =&gt; '1', 'Data' =&gt; { '1' =&gt; 'Yes', '0' =&gt; 'No' }, 'Colum' =&gt; '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-&gt;{'PreferencesGroups'}-&gt;{'LockTimeoutNotify'} = {   'Prio' =&gt; '3000',   'Label' =&gt; 'Ticket lock timeout notification',   'PrefKey' =&gt; 'UserSendLockTimeoutNotification',   'Module' =&gt; 'Kernel::Output::HTML::PreferencesGeneric',   'Desc' =&gt; 'Send me a notification if a ticket is unlocked by the system',   'Activ' =&gt; '1',   'Data' =&gt; {     '1' =&gt; 'Yes',     '0' =&gt; 'No'   },   'Colum' =&gt; '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-&gt;{'PreferencesGroups'}-&gt;{'MoveNotify'} = {   'Prio' =&gt; '4000',   'Label' =&gt; 'Move notification',   'PrefKey' =&gt; 'UserSendMoveNotification',   'Module' =&gt; 'Kernel::Output::HTML::PreferencesGeneric',   'Desc' =&gt; 'Send me a notification if a ticket is moved into one of "My',   'Activ' =&gt; '1',   'Data' =&gt; {     '1' =&gt; 'Yes',     '0' =&gt; 'No'   },   'Colum' =&gt; '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-&gt;{'PreferencesGroups'}-&gt;{'CustomQueue'} = { 'Prio' =&gt; '2000', 'Label' =&gt; 'My Queues', 'Module' =&gt; 'Kernel::Output::HTML::PreferencesCustomQueue', 'Desc' =&gt; 'Your queue selection of your favorite queues. You also get n 'Activ' =&gt; '1', 'Colum' =&gt; '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-&gt;{'PreferencesGroups'}-&gt;{'RefreshTime'} = { 'Prio' =&gt; '3000', 'Label' =&gt; 'QueueView refresh time', 'PrefKey' =&gt; 'UserRefreshTime', 'Module' =&gt; 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' =&gt; 'Select your QueueView refresh time.', 'Activ' =&gt; '1', 'Data' =&gt; { " =&gt; 'off', '7' =&gt; ' 7 minutes', '10' =&gt; '10 minutes', '2' =&gt; ' 2 minutes', '15' =&gt; '15 minutes', '5' =&gt; ' 5 minutes' }, 'Colum' =&gt; '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-&gt;{'PreferencesGroups'}-&gt;{'QueueView'} = { 'Prio' =&gt; '3000',          'Label' =&gt; 'QueueView', 'DataSelected' =&gt; 'AgentTicketQueueTicketView', 'PrefKey' =&gt; 'UserQueueView', 'Module' =&gt; 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' =&gt; 'Select your frontend QueueView.', 'Activ' =&gt; '1',           'Data' =&gt; { 'AgentTicketQueueTicketViewLite' =&gt; 'Lite', 'AgentTicketQueueTicketView' =&gt; 'Standard' }, 'Colum' =&gt; '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-&gt;{'PreferencesGroups'}-&gt;{'QueueViewShownTickets'} = { 'Prio' =&gt; '4000',          'Label' =&gt; 'Shown Tickets', 'DataSelected' =&gt; '15', 'PrefKey' =&gt; 'UserQueueViewShowTickets', 'Module' =&gt; 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' =&gt; 'Max. shown Tickets a page in QueueView.', 'Activ' =&gt; '1',           'Data' =&gt; { '25' =&gt; '25',              '10' =&gt; '10', '20' =&gt; '20',              '15' =&gt; '15' }, 'Colum' =&gt; '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-&gt;{'PreferencesGroups'}-&gt;{'CreateNextMask'} = { 'Prio' =&gt; '5000', 'Label' =&gt; 'Screen after new ticket', 'DataSelected' =&gt; "", 'PrefKey' =&gt; 'UserCreateNextMask', 'Module' =&gt; 'Kernel::Output::HTML::PreferencesGeneric', 'Desc' =&gt; 'Select your screen after creating a new ticket.', 'Activ' =&gt; '1',          'Data' =&gt; { " =&gt; 'CreateTicket', 'AgentTicketZoom' =&gt; 'TicketZoom' }, 'Colum' =&gt; '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-&gt;{'Ticket::Frontend::ArticleAttachmentModule'}-&gt;{'1-Download'} = 'Module' =&gt; '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-&gt;{'Ticket::Frontend::ArticleAttachmentModule'}-&gt;{'2-HTML-Viewer'} = 'Module' =&gt; '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-&gt;{'Ticket::Frontend::ArticleComposeModule'}-&gt;{'1-SignEmail'} = {   'Module' =&gt; '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-&gt;{'Ticket::Frontend::ArticleComposeModule'}-&gt;{'2-CryptEmail'} =   'Module' =&gt; '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-&gt;{'Ticket::Frontend::ArticleViewModule'}-&gt;{'1-PGP'} = { 'Module' =&gt; 'Kernel::Output::HTML::ArticleCheckPGP' };</pre>

#### B.2.14.2. Ticket::Frontend::ArticleViewModule###1-SMIME

Description	Value
Description:	Agent interface article notification module to check smime.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModule
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self-&gt;{'Ticket::Frontend::ArticleViewModule'}-&gt;{'1-SMIME'} = { 'Module' =&gt; '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-&gt;{'Ticket::Frontend::ArticlePreViewModule'}-&gt;{'1-PGP'} = { 'Module' =&gt; '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-&gt;{'Ticket::Frontend::ArticlePreViewModule'}-&gt;{'1-SMIME'} = {   'Module' =&gt; '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-&gt;{'Ticket::Frontend::MenuModule'}-&gt;{'000-Back'} = {   'Action' =&gt; "",   'Link' =&gt; '\$Env{"LastScreenOverview"}&amp;TicketID=\$QData{"TicketID"}',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Back',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::MenuModule'}-&gt;{'100-Lock'} = {   'Action' =&gt; 'AgentTicketLock',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuLock',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::MenuModule'}-&gt;{'200-History'} = {   'Action' =&gt; 'AgentTicketHistory',   'Link' =&gt; 'Action=AgentTicketHistory&amp;TicketID=\$QData{"TicketID"}',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Shows the ticket history!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::MenuModule'}-&gt;{'210-Print'} = {   'Action' =&gt; 'AgentTicketPrint',   'LinkParam' =&gt; 'target="print"',   'Link' =&gt; 'Action=AgentTicketPrint&amp;TicketID=\$QData{"TicketID"}',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Print this ticket!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::MenuModule'}-&gt;{'300-Priority'} = {   'Action' =&gt; 'AgentTicketPriority',   'Link' =&gt; 'Action=AgentTicketPriority&amp;TicketID=\$QData{"TicketID"}',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Change the ticket priority!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::MenuModule'}-&gt;{'310-FreeText'} = {   'Action' =&gt; 'AgentTicketFreeText',   'Link' =&gt; 'Action=AgentTicketFreeText&amp;TicketID=\$QData{"TicketID"}',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Change the ticket free fields!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::MenuModule'}-&gt;{'320-Link'} = {   'Action' =&gt; 'AgentLinkObject',   'Link' =&gt; 'Action=AgentLinkObject&amp;SourceObject=Ticket&amp;SourceID=\$QData{"   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Link this ticket to an other objects!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::MenuModule'}-&gt;{'400-Owner'} = {   'Action' =&gt; 'AgentTicketOwner',   'Link' =&gt; 'Action=AgentTicketOwner&amp;TicketID=\$QData{"TicketID"}',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Change the ticket owner!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::MenuModule'}-&gt;{'410-Customer'} = {   'Action' =&gt; 'AgentTicketCustomer',   'Link' =&gt; 'Action=AgentTicketCustomer&amp;TicketID=\$QData{"TicketID"}',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Change the ticket customer!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::MenuModule'}-&gt;{'420-Note'} = {   'Action' =&gt; 'AgentTicketNote',   'Link' =&gt; 'Action=AgentTicketNote&amp;TicketID=\$QData{"TicketID"}',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Add a note to this ticket!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::MenuModule'}-&gt;{'430-Merge'} = {   'Action' =&gt; 'AgentTicketMerge',   'Link' =&gt; 'Action=AgentTicketMerge&amp;TicketID=\$QData{"TicketID"}',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Merge this ticket!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::MenuModule'}-&gt;{'440-Pending'} = {   'Action' =&gt; 'AgentTicketPending',   'Link' =&gt; 'Action=AgentTicketPending&amp;TicketID=\$QData{"TicketID"}',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Set this ticket to pending!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::MenuModule'}-&gt;{'450-Close'} = {   'Action' =&gt; 'AgentTicketClose',   'Link' =&gt; 'Action=AgentTicketClose&amp;TicketID=\$QData{"TicketID"}',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Close this ticket!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::PreMenuModule'}-&gt;{'100-Lock'} = {   'Action' =&gt; 'AgentTicketLock',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuLock',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::PreMenuModule'}-&gt;{'200-Zoom'} = {   'Action' =&gt; 'AgentTicketZoom',   'Link' =&gt; 'Action=AgentTicketZoom&amp;TicketID=\$QData{"TicketID"}',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Look into a ticket!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::PreMenuModule'}-&gt;{'210-History'} = {   'Action' =&gt; 'AgentTicketHistory',   'Link' =&gt; 'Action=AgentTicketHistory&amp;TicketID=\$QData{"TicketID"}',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Shows the ticket history!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::PreMenuModule'}-&gt;{'300-Priority'} = {   'Action' =&gt; 'AgentTicketPriority',   'Link' =&gt; 'Action=AgentTicketPriority&amp;TicketID=\$QData{"TicketID"}',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Change the ticket priority!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::PreMenuModule'}-&gt;{'420-Note'} = {   'Action' =&gt; 'AgentTicketNote',   'Link' =&gt; 'Action=AgentTicketNote&amp;TicketID=\$QData{"TicketID"}',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Add a note to this ticket!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::PreMenuModule'}-&gt;{'440-Close'} = {   'Action' =&gt; 'AgentTicketClose',   'Link' =&gt; 'Action=AgentTicketClose&amp;TicketID=\$QData{"TicketID"}',   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Close this ticket!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::PreMenuModule'}-&gt;{'450-Delete'} = {   'Action' =&gt; 'AgentTicketMove',   'Link' =&gt; 'Action=AgentTicketMove&amp;TicketID=\$Data{"TicketID"}&amp;DestQueue=   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Delete this ticket!',   'Name' =&gt; '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-&gt;{'Ticket::Frontend::PreMenuModule'}-&gt;{'460-Spam'} = {   'Action' =&gt; 'AgentTicketMove',   'Link' =&gt; 'Action=AgentTicketMove&amp;TicketID=\$Data{"TicketID"}&amp;DestQueue=   'Module' =&gt; 'Kernel::Output::HTML::TicketMenuGeneric',   'Description' =&gt; 'Mark as Spam!',   'Name' =&gt; 'Spam' }; </pre>

**B.2.18. Frontend::Agent::Ticket::ViewBounce****B.2.18.1. Ticket::Frontend::BounceState**

Description	Value
Description:	Default bounce next state.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::BounceState'} = 'closed successful';</code>

### B.2.18.2. Ticket::DefaultNextBounceStateType

Description	Value
Description:	Next possible states for bounced message.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'Ticket::DefaultCloseNextStateType'} = [ 'closed' ];</code>

## B.2.20. Frontend::Agent::Ticket::ViewCompose

### B.2.20.1. Ticket::DefaultNextComposeType

Description	Value
Description:	Default compose next state.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::DefaultNextComposeType'} = 'open';</code>

### B.2.20.2. Ticket::DefaultNextComposeStateType

Description	Value
Description:	Next possible states for compose message.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self-&gt;{'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-&gt;{'Ticket::Frontend::ResponseFormat'} = '\$Data{"Salutation"} \$Data{"OrigFrom"} \$Text{"wrote"}: \$Data{"Body"} \$Data{"StdResponse"} \$Data{"Signature"} ';</pre>

## B.2.21. Frontend::Agent::Ticket::ViewEmailNew

### B.2.21.1. Ticket::Frontend::EmailPriority

Description	Value
Description:	Default priority for emails.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self-&gt;{'Ticket::Frontend::EmailPriority'} = '3 normal';</pre>

### B.2.21.2. Ticket::Frontend::EmailNewArticleType

Description	Value
Description:	Default article type for new article.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::EmailNewArticleType'} = 'email-external';</code>

### B.2.21.3. Ticket::Frontend::EmailNewSenderType

Description	Value
Description:	Default sender type for article.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::EmailNewSenderType'} = 'agent';</code>

### B.2.21.4. Ticket::Frontend::EmailNewHistoryType

Description	Value
Description:	Type for article history.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::EmailNewHistoryType'} = 'EmailAgent';</code>

### B.2.21.5. Ticket::Frontend::EmailNewHistoryComment

Description	Value
Description:	Default comment for new email history.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::EmailNewHistoryComment'} = "";</code>

#### B.2.21.6. Ticket::Frontend::EmailNewNote

Description	Value
Description:	Default note text for emails.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::EmailNewNote'} = "";</code>

#### B.2.21.7. Ticket::EmailDefaultNextStateType

Description	Value
Description:	Next possible states after e-mailphone.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::EmailDefaultNextStateType'} = [ 'open', 'pending auto', 'pending reminder', 'closed' ];</code>

#### B.2.21.8. Ticket::Frontend::EmailNewNextState

Description	Value
Description:	Default next state for the next new email.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'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-&gt;{'Ticket::Frontend::EmailNewShownCustomerTickets'} = '10';</code>

**B.2.22. Frontend::Agent::Ticket::ViewForward****B.2.22.1. Ticket::DefaultNextForwardStateType**

Description	Value
Description:	Next possible states for forwarded message.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'Ticket::Frontend::HistoryOrder'} = 'normal';</code>

**B.2.24. Frontend::Agent::Ticket::ViewMerge****B.2.24.1. Ticket::Frontend::MergeText**

Description	Value
Description:	Default merge text
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::MergeText'} = 'Your email with ticket number'</code>

## B.2.25. Frontend::Agent::Ticket::ViewMove

### B.2.25.1. 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-&gt;{'Ticket::Frontend::MoveType'} = 'form';</code>

### B.2.25.2. 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-&gt;{'Ticket::Frontend::MoveSetState'} = '1';</code>

### B.2.25.3. Ticket::DefaultNextMoveStateType

Description	Value
Description:	Possible state types after moving a ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1



Description	Value
Config-Setting:	<code>\$Self-&gt;{'Ticket::DefaultNextMoveStateType'} = [ 'open', 'closed' ];</code>

## B.2.26. Frontend::Agent::Ticket::ViewNote

### B.2.26.1. 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-&gt;{'Ticket::Frontend::NoteSetState'} = '0';</code>

### B.2.26.2. 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-&gt;{'Ticket::Frontend::NoteInformInvolvedAgent'} = '0';</code>

### B.2.26.3. 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-&gt;{'Ticket::Frontend::NoteInformAgent'} = '0';</code>

**B.2.26.4. Ticket::DefaultNextNoteStateType**

Description	Value
Description:	Default states of next note.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::DefaultNextNoteStateType'} = [ 'new', 'open', 'closed' ];</code>

**B.2.26.5. Ticket::Frontend::NoteType**

Description	Value
Description:	Default note type.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::NoteType'} = 'note-internal';</code>

**B.2.26.6. 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-&gt;{'Ticket::Frontend::NoteTypes'} = { 'note-external' =&gt; '0', 'note-internal' =&gt; '1', 'note-report' =&gt; '0' };</code>

**B.2.26.7. Ticket::Frontend::NoteSubject**

Description	Value
Description:	Default note subject.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::NoteSubject'} = '\$Text{"Note"}!';</code>

### B.2.26.8. Ticket::Frontend::NoteText

Description	Value
Description:	Default note text.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::NoteText'} = '';</code>

## B.2.27. Frontend::Agent::Ticket::ViewOwner

### B.2.27.1. 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-&gt;{'Ticket::Frontend::OwnerSetState'} = '0';</code>

### B.2.27.2. Ticket::DefaultNextOwnerStateType

Description	Value
Description:	Possible states of next owner screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::DefaultNextOwnerStateType'} = [ 'open', 'closed' ];</code>

### B.2.27.3. 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-&gt;{'Ticket::Frontend::OwnerSubject'} = '\$Text{"Owner Update"}';</code>

## B.2.28. Frontend::Agent::Ticket::ViewPending

### B.2.28.1. 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-&gt;{'Ticket::Frontend::PendingSubject'} = '\$Text{"Pending"}!';</code>

### B.2.28.2. Ticket::Frontend::PendingText

Description	Value
Description:	Default text for pending tickets.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::PendingText'} = '';</code>

### B.2.28.3. 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-&gt;{'Ticket::Frontend::PendingState'} = 'pending reminder';</code>

### B.2.28.4. 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-&gt;{'Ticket::DefaultPendingNextStateType'} = [ 'pending reminder', 'pending auto' ];</code>

## B.2.29. Frontend::Agent::Ticket::ViewPhone

### B.2.29.1. Ticket::Frontend::PhoneArticleType

Description	Value
Description:	Default note type for phone tickets.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewPhone
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::PhoneArticleType'} = 'phone';</code>

### B.2.29.2. 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-&gt;{'Ticket::Frontend::PhoneSenderType'} = 'agent';</code>

### B.2.29.3. 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-&gt;{'Ticket::Frontend::PhoneSubject'} = '\$Text{"Phone call"}!';</code>

### B.2.29.4. Ticket::Frontend::PhoneNote

Description	Value
Description:	Default note text for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhone
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::PhoneNote'} = 'Customer called';</code>

#### B.2.29.5. Ticket::PhoneDefaultNextStateType

Description	Value
Description:	Next possible states after phone.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhone
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::PhoneDefaultNextStateType'} = [ 'open', 'pending auto', 'pending reminder', 'closed' ];</code>

#### B.2.29.6. Ticket::Frontend::PhoneNextState

Description	Value
Description:	Default next state after phone.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhone
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::PhoneNextState'} = 'closed successful';</code>

#### B.2.29.7. Ticket::Frontend::PhoneHistoryType

Description	Value
Description:	Default history type for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhone
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::PhoneHistoryType'} = 'PhoneCallAgent';</code>

### B.2.29.8. 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-&gt;{'Ticket::Frontend::PhoneHistoryComment'} = '';</code>

### B.2.29.9. Ticket::Frontend::PhonePriority

Description	Value
Description:	Default priority for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhone
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::PhonePriority'} = '3 normal';</code>

## B.2.30. Frontend::Agent::Ticket::ViewPhoneNew

### B.2.30.1. Ticket::Frontend::PhoneNewArticleType

Description	Value
Description:	Default article type for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1



Description	Value
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::PhoneNewArticleType'} = 'phone';</code>

### B.2.30.2. 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-&gt;{'Ticket::Frontend::PhoneNewSenderType'} = 'customer';</code>

### B.2.30.3. Ticket::Frontend::PhoneNewSubject

Description	Value
Description:	Default subject for new phone tickets (e.g. 'Phone call').
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::PhoneNewSubject'} = '';</code>

### B.2.30.4. Ticket::Frontend::PhoneNewNote

Description	Value
Description:	Default note text. E.g 'New ticket via call.'
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::PhoneNewNote'} = '';</code>

**B.2.30.5. Ticket::Frontend::PhoneNewNextState**

Description	Value
Description:	Default next state.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::PhoneNewNextState'} = 'open';</code>

**B.2.30.6. 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-&gt;{'Ticket::Frontend::PhoneNewShownCustomerTickets'} = '10';</code>

**B.2.30.7. Ticket::Frontend::PhoneNewHistoryType**

Description	Value
Description:	Default history type for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::PhoneNewHistoryType'} = 'PhoneCallCustomer';</code>

**B.2.30.8. Ticket::Frontend::PhoneNewHistoryComment**

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

Description	Value
Description:	Default history comment for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::PhoneNewHistoryComment'} = '';</code>

## B.2.31. Frontend::Agent::Ticket::ViewQueue

### B.2.31.1. 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-&gt;{'Ticket::QueueViewAllPossibleTickets'} = '0';</code>

### B.2.31.2. HighlightAge1

Description	Value
Description:	Set the age (first level) for highlighting of old queue (Age in min).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'HighlightAge1'} = '1440';</code>

### B.2.31.3. HighlightColor1

Description	Value
Description:	Set the color for highlighting of old queue (first level).

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'HighlightColor1'} = 'orange';</code>

#### B.2.31.4. HighlightAge2

Description	Value
Description:	Set the age (second level) for highlighting of old queue (Age in min).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'HighlightAge2'} = '2880';</code>

#### B.2.31.5. 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-&gt;{'HighlightColor2'} = 'red';</code>

#### B.2.31.6. Ticket::Frontend::NoEscalationGroup

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

Description	Value
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::NoEscalationGroup'} = 'some_group';</code>

### B.2.31.7. 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-&gt;{'Ticket::Frontend::QueueMaxShown'} = '1200';</code>

### B.2.31.8. Ticket::Frontend::QueueSortDefault

Description	Value
Description:	Default sort order of the queue view (after priority sort). ASC: oldest on top. DESC: youngest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::QueueSortDefault'} = 'ASC';</code>

### B.2.31.9. Ticket::Frontend::QueueSort

Description	Value
Description:	Sort a queue ascending or descending after priority sort. Values: 0 = ascending (oldest on top, default), 1 = descending (youngest on top)
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	0
Required:	0
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::QueueSort'} = { '7' =&gt; '1', '3' =&gt; '0'};</code>

## B.2.32. Frontend::Agent::Ticket::ViewSearch

### B.2.32.1. 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-&gt;{'Ticket::Frontend::SearchLimit'} = '5000';</code>

### B.2.32.2. Ticket::Frontend::SearchPageShown

Description	Value
Description:	Default of shown article a page.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::SearchPageShown'} = '40';</code>

### B.2.32.3. 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-&gt;{'Ticket::Frontend::SearchViewableTicketLines'} = '10';</code>

### B.2.32.4. Ticket::Frontend::SortBy::Default

Description	Value
Description:	Search result sort by default.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::SortBy::Default'} = 'Age';</code>

#### B.2.32.5. Ticket::Frontend::Order::Default

Description	Value
Description:	Search result order default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::Order::Default'} = 'Down';</code>

#### B.2.32.6. 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-&gt;{'Ticket::Frontend::SearchArticleCSVTree'} = '0';</code>

#### B.2.32.7. 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

Description	Value
Required:	1
Config-Setting:	<pre>\$Self-&gt;{'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.2.32.8. 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-&gt;{'Ticket::Frontend::Search::DB::DSN'} = 'DBI:mysql:database=mir</pre>

#### B.2.32.9. 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-&gt;{'Ticket::Frontend::Search::DB::User'} = 'some_user';</code>

### B.2.32.10. 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-&gt;{'Ticket::Frontend::Search::DB::Password'} = 'some_password';</code>

## B.2.33. Frontend::Agent::Ticket::ViewStatus

### B.2.33.1. 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-&gt;{'Ticket::Frontend::StatusView::ViewableTicketsPage'} = '50';</code>

## B.2.34. Frontend::Agent::Ticket::ViewZoom

### B.2.34.1. Ticket::Frontend::ZoomExpand

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

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-&gt;{'Ticket::Frontend::ZoomExpand'} = '0';</code>

#### B.2.34.2. Ticket::Frontend::ZoomExpandSort

Description	Value
Description:	Show article normal or in reverse order in the agent interface under ticket zoom.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::Frontend::ZoomExpandSort'} = 'normal';</code>

#### B.2.34.3. 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-&gt;{'Ticket::ZoomAttachmentDisplay'} = '1';</code>

#### B.2.34.4. 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

Description	Value
Required:	0
Config-Setting:	<code>\$Self-&gt;{'Ticket::ZoomTimeDisplay'} = '0';</code>

## B.2.35. Frontend::Customer

### B.2.35.1. CustomerPriority

Description	Value
Description:	Is the customer allowed to set the priority for a ticket?
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'CustomerPriority'} = '1';</code>

### B.2.35.2. CustomerDefaultPriority

Description	Value
Description:	Default priority of new customer tickets.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'CustomerDefaultPriority'} = '3 normal';</code>

### B.2.35.3. CustomerDefaultState

Description	Value
Description:	Ddefault state of new customer tickets.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'CustomerDefaultState'} = 'new';</code>

**B.2.35.4. CustomerNextScreenAfterNewTicket**

Description	Value
Description:	Next screen after new customer ticket.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'CustomerNextScreenAfterNewTicket'} = 'CustomerTicketOverview'</code>

**B.2.35.5. CustomerPanelArticleType**

Description	Value
Description:	Type of note.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'CustomerPanelArticleType'} = 'webrequest';</code>

**B.2.35.6. 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-&gt;{'CustomerPanelSenderType'} = 'customer';</code>

**B.2.35.7. CustomerPanelHistoryType**

Description	Value
Description:	Default history type in the customer panel.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self-&gt;{'CustomerPanelHistoryType'} = 'FollowUp';</code>

### B.2.35.8. CustomerPanelHistoryComment

Description	Value
Description:	Default history comment in the customer panel.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'CustomerPanelHistoryComment'} = '';</code>

### B.2.35.9. CustomerPanelDefaultNextComposeType

Description	Value
Description:	Next state for ticket after customer followup.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'CustomerPanelDefaultNextComposeType'} = 'open';</code>

### B.2.35.10. CustomerPanelNextComposeState

Description	Value
Description:	Next commpose state for customer tickets.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'CustomerPanelNextComposeState'} = '1';</code>

**B.2.35.11. Ticket::CustomerPanelDefaultNextComposeStateType**

Description	Value
Description:	Next possible states for customer tickets.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::CustomerPanelDefaultNextComposeStateType'} = [ 'open', 'closed' ];</code>

**B.2.35.12. CustomerPanelNewArticleType**

Description	Value
Description:	Default article type.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'CustomerPanelNewArticleType'} = 'webrequest';</code>

**B.2.35.13. 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-&gt;{'CustomerPanelNewSenderType'} = 'customer';</code>

**B.2.35.14. CustomerPanelNewHistoryType**

Description	Value
Description:	Default history type.
Group:	Ticket
SubGroup:	Frontend::Customer

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'CustomerPanelNewHistoryType'} = 'WebRequestCustomer';</code>

#### B.2.35.15. 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-&gt;{'CustomerPanelNewHistoryComment'} = '';</code>

#### B.2.35.16. 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-&gt;{'CustomerPanelSelectionType'} = 'Queue';</code>

#### B.2.35.17. 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-&gt;{'CustomerPanelSelectionString'} = '&lt;Queue&gt;';</code>

**B.2.35.18. CustomerPanelOwnSelection**

Description	Value
Description:	CustomerPanelOwnSelection. If this is in use, just this selection is valid" for the CustomMessage.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self-&gt;{'CustomerPanelOwnSelection'} = { 'Misc' =&gt; 'Second Queue', 'Junk' =&gt; 'First Queue' };</pre>

**B.2.35.19. CustomerPanel::NewTicketQueueSelectionModule**

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:	<pre>\$Self-&gt;{'CustomerPanel::NewTicketQueueSelectionModule'} =</pre>

'Kernel::Out

**B.2.35.20. Ticket::CustomerTicketSearch::SearchLimit**

Description	Value
Description:	Default limit for ticket search.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self-&gt;{'Ticket::CustomerTicketSearch::SearchLimit'} =</pre>

'5000';

**B.2.35.21. Ticket::CustomerTicketSearch::SearchPageShown**

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



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-&gt;{'Ticket::CustomerTicketSearch::SearchPageShown'} = '40';</code>

#### B.2.35.22. Ticket::CustomerTicketSearch::SortBy::Default

Description	Value
Description:	Search result sort by default.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::CustomerTicketSearch::SortBy::Default'} = 'Age';</code>

#### B.2.35.23. Ticket::CustomerTicketSearch::Order::Default

Description	Value
Description:	Search result order default.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'Ticket::CustomerTicketSearch::Order::Default'} = 'Down';</code>

#### B.2.35.24. CustomerFrontend::CommonObject###QueueObject

Description	Value
Description:	Path to the file where all settings for the QueueObject object for the customer interface are stored.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'CustomerFrontend::CommonObject'}-&gt;{'QueueObject'} = 'Kernel::</code>

#### B.2.35.25. 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
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'CustomerFrontend::CommonObject'}-&gt;{'TicketObject'} = 'Kernel::</code>

#### B.2.35.26. CustomerFrontend::CommonParam###Action

Description	Value
Description:	Standardwert für den Action-Parameter innerhalb des Customer-Interfaces.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'CustomerFrontend::CommonParam'}-&gt;{'Action'} = 'CustomerTicket</code>

#### B.2.35.27. CustomerFrontend::CommonParam###QueueID

Description	Value
Description:	Standard parameter for the QueueID in the customer interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'CustomerFrontend::CommonParam'}-&gt;{'QueueID'} = '0';</code>

### B.2.35.28. CustomerFrontend::CommonParam###TicketID

Description	Value
Description:	Standard parameter for the TicketID in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-&gt;{'CustomerFrontend::CommonParam'}-&gt;{'TicketID'} = '';</code>

## B.2.36. Frontend::Customer::ModuleRegistration

### B.2.36.1. 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

Description	Value
Config-Setting:	<pre> \$Self-&gt;{'CustomerFrontend::Module'}-&gt;{'CustomerTicketOverView'} = {   'NavBar' =&gt; [     {       'Prio' =&gt; '110',       'Block' =&gt; "",       'Image' =&gt; 'ticket.png',       'NavBar' =&gt; "",       'Type' =&gt; "",       'AccessKey' =&gt; 'm',       'Description' =&gt; 'MyTickets',       'Name' =&gt; 'MyTickets',       'Link' =&gt; 'Action=CustomerTicketOverView&amp;Type=MyTickets'     },     {       'Prio' =&gt; '120',       'Block' =&gt; "",       'Image' =&gt; 'tickets.png',       'NavBar' =&gt; "",       'Type' =&gt; "",       'AccessKey' =&gt; 'c',       'Description' =&gt; 'CompanyTickets',       'Name' =&gt; 'CompanyTickets',       'Link' =&gt; 'Action=CustomerTicketOverView&amp;Type=CompanyTickets'     }   ],   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Overview of customer tickets',   'Title' =&gt; 'Overview' }; </pre>

### B.2.36.2. 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

Description	Value
Config-Setting:	<pre> \$Self-&gt;{'CustomerFrontend::Module'}-&gt;{'CustomerTicketMessage'} = {   'NavBar' =&gt; [     {       'Prio' =&gt; '100',       'Block' =&gt; "",       'Image' =&gt; 'new.png',       'NavBar' =&gt; "",       'Type' =&gt; "",       'AccessKey' =&gt; 'n',       'Description' =&gt; 'Create new Ticket',       'Name' =&gt; 'New Ticket',       'Link' =&gt; 'Action=CustomerTicketMessage'     }   ],   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Create and updated tickets',   'Title' =&gt; 'Message' }; </pre>

### B.2.36.3. 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-&gt;{'CustomerFrontend::Module'}-&gt;{'CustomerTicketZoom'} = {   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Ticket zoom view',   'Title' =&gt; 'Zoom' }; </pre>

### B.2.36.4. CustomerFrontend::Module###CustomerZoom

Description	Value
Description:	Frontend module registration for the CustomerZoom object in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self-&gt;{'CustomerFrontend::Module'}-&gt;{'CustomerZoom'} = {   'NavBarName' =&gt; " ,   'Description' =&gt; 'compat mod' ,           'Title' =&gt; " };</pre>

#### B.2.36.5. 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-&gt;{'CustomerFrontend::Module'}-&gt;{'CustomerTicketAttachment'} = {   'NavBarName' =&gt; " ,   'Description' =&gt; 'To download attachments' ,   'Title' =&gt; "           }; </pre>

#### B.2.36.6. 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-&gt;{'CustomerFrontend::Module'}-&gt;{'CustomerTicketSearch'} = {   'NavBar' =&gt; [     {       'Prio' =&gt; '300',       'Block' =&gt; "",       'Image' =&gt; 'search.png',       'NavBar' =&gt; "",       'Type' =&gt; "",       'AccessKey' =&gt; 's',       'Description' =&gt; 'Search',       'Name' =&gt; 'Search',       'Link' =&gt; 'Action=CustomerTicketSearch'     }   ],   'NavBarName' =&gt; 'Ticket',   'Description' =&gt; 'Customer ticket search',   'Title' =&gt; 'Search' }; </pre>

## B.2.37. Frontend::Customer::Preferences

### B.2.37.1. 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-&gt;{'CustomerPreferencesGroups'}-&gt;{'ClosedTickets'} = {   'Prio' =&gt; '2000',   'Label' =&gt; 'Closed Tickets',   'DataSelected' =&gt; '1',   'PrefKey' =&gt; 'UserShowClosedTickets',   'Module' =&gt; 'Kernel::Output::HTML::PreferencesGeneric',   'Desc' =&gt; 'Show closed tickets.',   'Activ' =&gt; '1',   'Data' =&gt; {     '1' =&gt; 'Yes',     '0' =&gt; 'No'   },   'Colum' =&gt; 'Other Options' }; </pre>

### B.2.37.2. 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-&gt;{'CustomerPreferencesGroups'}-&gt;{'ShownTickets'} = {   'Prio' =&gt; '4000',          'Label' =&gt; 'Shown Tickets',   'DataSelected' =&gt; '25',   'PrefKey' =&gt; 'UserShowTickets',   'Module' =&gt; 'Kernel::Output::HTML::PreferencesGeneric',   'Desc' =&gt; 'Max. shown Tickets a page in Overview.',   'Activ' =&gt; '1',           'Data' =&gt; {     '25' =&gt; '25',           '30' =&gt; '30',     '20' =&gt; '20',           '15' =&gt; '15'   },   'Colum' =&gt; 'Frontend'    }; </pre>

### B.2.37.3. 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-&gt;{'CustomerPreferencesGroups'}-&gt;{'RefreshTime'} = {   'Prio' =&gt; '4000',   'Label' =&gt; 'QueueView refresh time',   'PrefKey' =&gt; 'UserRefreshTime',   'Module' =&gt; 'Kernel::Output::HTML::PreferencesGeneric',   'Desc' =&gt; 'Select your QueueView refresh time.',   'Activ' =&gt; '1',           'Data' =&gt; {     '' =&gt; 'off',           '7' =&gt; '7 minutes',     '10' =&gt; '10 minutes',     '2' =&gt; '2 minutes',     '15' =&gt; '15 minutes',     '5' =&gt; '5 minutes'   },   'Colum' =&gt; '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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'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-&gt;{'LinkObject'}-&gt;{'FAQ'} = {  'Type' =&gt; 'Object',                    'LinkObjects' =&gt; [  'Ticket',                                'FAQ'  ],                                        'Name' =&gt; '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-&gt;{'Frontend::Module'}-&gt;{'FAQ'} = { 'NavBar' =&gt; [ 'Prio' =&gt; '8300', 'Block' =&gt; 'ItemArea', 'GroupRo' =&gt; [ 'faq' ], 'Image' =&gt; 'help.png', 'NavBar' =&gt; 'FAQ', 'Type' =&gt; 'Menu', 'AccessKey' =&gt; 'q', 'Description' =&gt; 'FAQ-Area', 'Name' =&gt; 'FAQ', 'Link' =&gt; 'Action=FAQ&amp;Nav=Normal' }, 'Prio' =&gt; '200', 'Block' =&gt; "", 'Image' =&gt; 'new.png', 'Group' =&gt; [ 'faq' ], 'NavBar' =&gt; 'FAQ', 'Type' =&gt; "", 'AccessKey' =&gt; 'n', 'Description' =&gt; 'New Article', 'Name' =&gt; 'New Article', 'Link' =&gt; 'Action=FAQ&amp;Subaction=Add' }, 'Prio' =&gt; '300', 'Block' =&gt; "", 'GroupRo' =&gt; [ 'faq' ], 'Image' =&gt; 'search.png', 'NavBar' =&gt; 'FAQ', 'Type' =&gt; "", 'AccessKey' =&gt; 's', 'Description' =&gt; 'FAQ-Search', 'Name' =&gt; 'Search', 'Link' =&gt; 'Action=FAQ&amp;Subaction=Search' }, 'Prio' =&gt; '310', 'Block' =&gt; "", 'GroupRo' =&gt; [ 'faq' ], 'Image' =&gt; 'list.png', 'NavBar' =&gt; 'FAQ', 'Type' =&gt; "", 'AccessKey' =&gt; 'o', 'Description' =&gt; 'History', 'Name' =&gt; 'History', 'Link' =&gt; 'Action=FAQ&amp;Subaction=SystemHistory' } ], 'GroupRo' =&gt; [ 'faq' ], 'NavBarName' =&gt; 'FAQ', 'Description' =&gt; 'FAQ-Area', 'Title' =&gt; "", 'Group' =&gt; [ '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-&gt;{'Frontend::Module'}-&gt;{'FAQCategory'} = { 'NavBar' =&gt; [ 'Prio' =&gt; '900', 'Image' =&gt; 'fileopen.png', 'NavBar' =&gt; 'FAQ', 'AccessKey' =&gt; 'g', 'Description' =&gt; 'Category', 'Name' =&gt; 'Category', 'Link' =&gt; 'Action=FAQCategory' ], 'GroupRo' =&gt; [ ], 'NavBarName' =&gt; 'FAQ', 'Description' =&gt; 'FAQ-Category', 'Title' =&gt; 'Category', 'faq' =&gt; [ ] ]; </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-&gt;{'Frontend::Module'}-&gt;{'FAQLanguage'} = {   'NavBar' =&gt; [     {       'Prio' =&gt; '910',       'Block' =&gt; "",       'Image' =&gt; 'fileopen.png',       'NavBar' =&gt; 'FAQ',       'Type' =&gt; "",       'AccessKey' =&gt; 'u',       'Description' =&gt; 'Language',       'Name' =&gt; 'Language',       'Link' =&gt; 'Action=FAQLanguage'     }   ],   'GroupRo' =&gt; [     "",     'NavBarName' =&gt; 'FAQ',     'Description' =&gt; 'FAQ-Language',     'Title' =&gt; 'Language',     'Group' =&gt; [       '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-&gt;{'CustomerFrontend::Module'}-&gt;{'CustomerFAQ'} = {   'NavBar' =&gt; [     {       'Prio' =&gt; '400',       'Block' =&gt; "",       'Image' =&gt; 'help.png',       'NavBar' =&gt; "",       'Type' =&gt; "",       'AccessKey' =&gt; 'f',       'Description' =&gt; 'FAQ-Area',       'Name' =&gt; 'FAQ-Area',       'Link' =&gt; 'Action=CustomerFAQ'     }   ],   'NavBarName' =&gt; 'FAQ',   'Description' =&gt; 'Customer faq',   'Title' =&gt; "" }; </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-&gt;{'PublicFrontend::CommonParam'}-&gt;{'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-&gt;{'PublicFrontend::Module'}-&gt;{'PublicFAQ'} = {  'NavBar' =&gt; [  'Prio' =&gt; '400',  'Image' =&gt; 'help.png',  'NavBar' =&gt; "",  'AccessKey' =&gt; 'f',  'Description' =&gt; 'FAQ-Area',  'Name' =&gt; 'FAQ-Area',  'Link' =&gt; 'Action=CustomerFAQ'  ],  'NavBarName' =&gt; 'FAQ',  'Description' =&gt; '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 forgotten someone definitely, 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 Wwin32-installer for the 1.x releases. Thanks a lot, Robert!
- Torsten Werner, who maintains the Debian-installer for OTRS. Thanks a lot, Torsten, to make 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!
- Via the OTRS mailinglists we receive many ideas for system improvements, patches or bugfixes and great support for all users is available. Thanks a lot to all people on the mailinglists for your active assistance!

Also we want to give a big thank 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.