

OTRS 3.0 - Admin Manual

OTRS 3.0 - Admin Manual

Cancún Beach (México) Edition

Copyright © 2003-2010 OTRS AG

René Bakker, Stefan Bedorf, Michiel Beijen, Shawn Beasley, Hauke Böttcher, Jens Bothe, Udo Bretz, Martin Edenhofer, Carlos Javier García, Martin Gruner, Manuel Hecht, Christopher Kuhn, André Mindermann, Marc Nilius, Elva María Novoa, Henning Oschwald, Martha Elia Pascual, Thomas Raith, Carlos Fernando Rodríguez, Stefan Rother, Burchard Steinbild, Daniel Zamorano.

This work is copyrighted by OTRS AG.

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, Windows 2003, Windows Vista and Windows 7 are registered trademarks of Microsoft Corporation. Other trademarks and registered trademarks are: SUSE and YaST of SUSE Linux 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. MySQL and the MySQL Logo are registered trademarks of MySQL AB.

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

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

Table of Contents

Preface	??
1. Basics about trouble ticket systems.....	??
1.1. What is a trouble ticket system and why do you need it?	??
1.2. What is a trouble ticket?.....	??
2. OTRS - Open Ticket Request System	??
2.1. Basics	??
2.2. Features	??
2.2.1. Top new features of OTRS 3.0	??
2.2.2. New features of OTRS 2.4.....	??
2.2.3. New features of OTRS 2.3.....	??
2.2.4. New features of OTRS 2.2.....	??
2.2.5. New features of OTRS 2.1.....	??
2.3. Hardware and software requirements.....	??
2.4. Community.....	??
2.5. Commercial Support for OTRS.....	??
3. Installation / Upgrade of the OTRS framework	??
3.1. The simple way - Installation of pre-built packages	??
3.1.1. Installing the rpm on a SUSE Linux server.....	??
3.1.2. Installing OTRS on a CentOS system	??
3.1.3. Installing OTRS on a Debian system	??
3.1.4. Installing OTRS on a Ubuntu system	??
3.1.5. Installing OTRS on Microsoft Windows systems	??
3.2. Installation from source (Linux, Unix)	??
3.2.1. Preparing the installation from source.....	??
3.2.2. Installation of Perl modules.....	??
3.2.3. Configuring the apache web server	??
3.2.4. Configuring the database	??
3.2.5. Setting up the cron jobs for OTRS	??
3.3. Upgrading the OTRS Framework	??
3.4. Upgrading Windows Installer	??
4. First steps in OTRS.....	??
4.1. Agent web interface	??
4.2. Customer web interface.....	??
4.3. Public web interface.....	??
4.4. First login	??
4.5. An overview to the web interface.....	??
4.6. What is a queue?	??
4.7. User preferences.....	??
5. The admin area of OTRS	??
5.1. Basics	??
5.2. Agents, groups and roles	??
5.2.1. Agents.....	??
5.2.2. Groups	??
5.2.3. Roles	??

5.3. Customers and customer groups	??
5.3.1. Customers	??
5.3.2. Customer groups.....	??
5.4. Queues.....	??
5.5. Salutations, signatures, attachments and responses	??
5.5.1. Salutations	??
5.5.2. Signatures	??
5.5.3. Attachments.....	??
5.5.4. Responses	??
5.6. Auto responses	??
5.7. Email addresses.....	??
5.8. Notifications.....	??
5.9. SMIME.....	??
5.10. PGP	??
5.11. States	??
5.12. SysConfig	??
5.13. Using mail accounts	??
5.14. Filtering incoming messages.....	??
5.15. Executing automated jobs with the GenericAgent.....	??
5.16. Admin email.....	??
5.17. Session management	??
5.18. System Log	??
5.19. SQL queries via the SQL box	??
5.20. Package manager.....	??
6. Configuration of the system	??
6.1. The config files of OTRS	??
6.2. Configuring the system through the web interface	??
7. Sending/Receiving emails.....	??
7.1. Sending emails	??
7.1.1. Via Sendmail (default).....	??
7.1.2. Via SMTP server or smarthost.....	??
7.2. Receiving emails	??
7.2.1. Mail accounts configured via the OTRS GUI	??
7.2.2. Via command line program and e.g. procmail (otrs.PostMaster.pl).....	??
7.2.3. Fetching emails via POP3 or IMAP and fetchmail for otrs.PostMaster.pl.....	??
7.2.4. Filtering/dispatching by OTRS/PostMaster modules (for more complex dispatching)??	
8. Time related functions.....	??
8.1. Setting up business hours, holidays and time zones	??
8.1.1. Business Hours	??
8.1.2. Fixed date holidays.....	??
8.1.3. TimeVacationDaysOneTime.....	??
8.2. Automated Unlocking	??
9. Ticket responsibility and ticket watching	??
9.1. Ticket responsibility	??
9.2. Ticket watching	??

10. Customize the PDF output.....	??
11. Using external backends.....	??
11.1. Customer data	??
11.2. Customer user backend	??
11.2.1. Database (Default).....	??
11.2.2. LDAP	??
11.2.3. Use more than one customer backend with OTRS	??
11.3. Backends to authenticate agents and customers.....	??
11.3.1. Authentication backends for agents.....	??
11.3.2. Authentication backends for customers.....	??
11.4. Customize the customer self registration	??
11.4.1. Customizing the web interface	??
11.4.2. Customer mapping.....	??
11.4.3. Customize the customer_user table in the OTRS DB	??
12. States	??
12.1. Predefined states.....	??
12.1.1. New.....	??
12.1.2. Open	??
12.1.3. Pending reminder.....	??
12.1.4. Pending auto close-.....	??
12.1.5. Pending auto close+.....	??
12.1.6. Merged.....	??
12.1.7. Closed Successful	??
12.1.8. Closed Unsuccessful.....	??
12.2. Customizing states	??
13. Modifying ticket priorities	??
14. Creating your own themes	??
15. Localization of the OTRS frontend	??
16. PGP	??
17. S/MIME	??
18. Access Control Lists (ACLs)	??
19. Stats module	??
19.1. Handling of the module by the agent	??
19.1.1. Overview	??
19.1.2. Generate and view stats	??
19.1.3. Edit - New.....	??
19.1.4. Import	??
19.2. Administration of the stat module by the OTRS administrator	??
19.2.1. Permission settings, groups and queues	??
19.2.2. SysConfig	??
19.3. Administration of the stats module by the system administrator	??
19.3.1. Data base table.....	??
19.3.2. List of all files	??
19.3.3. Caching.....	??
19.3.4. otrs.GenerateStats.pl.....	??
19.3.5. Automated stat generation - Cronjob	??

19.3.6. Static stats	??
19.3.7. Using old static stats	??
19.3.8. Default stats	??
20. Additional applications	??
20.1. Calendar	??
20.2. File Manager	??
20.3. Web Mail	??
20.4. FAQ	??
21. Performance Tuning	??
21.1. OTRS	??
21.1.1. TicketIndexModule	??
21.1.2. TicketStorageModule	??
21.1.3. Archiving Tickets	??
21.2. Database	??
21.2.1. MySQL	??
21.2.2. PostgreSQL	??
21.3. Webserver	??
21.3.1. Pre-establish database connections	??
21.3.2. Preloaded modules - startup.pl	??
21.3.3. Reload Perl modules when updated on disk	??
21.3.4. Choosing the Right Strategy	??
21.3.5. mod_gzip/mod_deflate	??
22. Backing up the system	??
22.1. Back up	??
22.2. Restore	??
A. Additional Resources	??
A.1. Homepage OTRS.org	??
A.2. Mailing lists	??
A.3. Bug tracking	??
A.4. Commercial Support	??
B. Configuration Options Reference	??
B.1. Framework	??
B.1.1. Core	??
B.1.2. Core::LinkObject	??
B.1.3. Core::Log	??
B.1.4. Core::MIME-Viewer	??
B.1.5. Core::MirrorDB	??
B.1.6. Core::PDF	??
B.1.7. Core::Package	??
B.1.8. Core::PerformanceLog	??
B.1.9. Core::SOAP	??
B.1.10. Core::Sendmail	??
B.1.11. Core::Session	??
B.1.12. Core::SpellChecker	??
B.1.13. Core::Stats	??
B.1.14. Core::Stats::Graph	??

B.1.15. Core::Time	??
B.1.16. Core::Time::Calendar1	??
B.1.17. Core::Time::Calendar2	??
B.1.18. Core::Time::Calendar3	??
B.1.19. Core::Time::Calendar4	??
B.1.20. Core::Time::Calendar5	??
B.1.21. Core::Time::Calendar6	??
B.1.22. Core::Time::Calendar7	??
B.1.23. Core::Time::Calendar8	??
B.1.24. Core::Time::Calendar9	??
B.1.25. Core::Web	??
B.1.26. Core::WebUserAgent	??
B.1.27. Crypt::PGP	??
B.1.28. Crypt::SMIME	??
B.1.29. Frontend::Admin::ModuleRegistration	??
B.1.30. Frontend::Agent	??
B.1.31. Frontend::Agent::Dashboard	??
B.1.32. Frontend::Agent::ModuleMetaHead	??
B.1.33. Frontend::Agent::ModuleNotify	??
B.1.34. Frontend::Agent::ModuleRegistration	??
B.1.35. Frontend::Agent::NavBarModule	??
B.1.36. Frontend::Agent::Preferences	??
B.1.37. Frontend::Agent::SearchRouter	??
B.1.38. Frontend::Agent::Stats	??
B.1.39. Frontend::Customer	??
B.1.40. Frontend::Customer::Auth	??
B.1.41. Frontend::Customer::ModuleMetaHead	??
B.1.42. Frontend::Customer::ModuleNotify	??
B.1.43. Frontend::Customer::ModuleRegistration	??
B.1.44. Frontend::Customer::Preferences	??
B.1.45. Frontend::Public	??
B.1.46. Frontend::Public::ModuleRegistration	??
B.2. Ticket	??
B.2.1. Core::ArticleFreeText	??
B.2.2. Core::FulltextSearch	??
B.2.3. Core::LinkObject	??
B.2.4. Core::PostMaster	??
B.2.5. Core::Stats	??
B.2.6. Core::Ticket	??
B.2.7. Core::TicketACL	??
B.2.8. Core::TicketBulkAction	??
B.2.9. Core::TicketFreeText	??
B.2.10. Core::TicketFreeTextDefault	??
B.2.11. Core::TicketFreeTime	??
B.2.12. Core::TicketWatcher	??
B.2.13. Frontend::Admin::ModuleRegistration	??
B.2.14. Frontend::Agent	??
B.2.15. Frontend::Agent::CustomerSearch	??

B.2.16. Frontend::Agent::Dashboard	??
B.2.17. Frontend::Agent::ModuleMetaHead.....	??
B.2.18. Frontend::Agent::ModuleNotify	??
B.2.19. Frontend::Agent::ModuleRegistration.....	??
B.2.20. Frontend::Agent::Preferences	??
B.2.21. Frontend::Agent::SearchRouter	??
B.2.22. Frontend::Agent::Ticket::ArticleAttachmentModule	??
B.2.23. Frontend::Agent::Ticket::ArticleComposeModule.....	??
B.2.24. Frontend::Agent::Ticket::ArticleViewModule.....	??
B.2.25. Frontend::Agent::Ticket::ArticleViewModulePre	??
B.2.26. Frontend::Agent::Ticket::MenuModule.....	??
B.2.27. Frontend::Agent::Ticket::MenuModulePre	??
B.2.28. Frontend::Agent::Ticket::ViewBounce	??
B.2.29. Frontend::Agent::Ticket::ViewBulk	??
B.2.30. Frontend::Agent::Ticket::ViewClose	??
B.2.31. Frontend::Agent::Ticket::ViewCompose	??
B.2.32. Frontend::Agent::Ticket::ViewCustomer.....	??
B.2.33. Frontend::Agent::Ticket::ViewEmailNew	??
B.2.34. Frontend::Agent::Ticket::ViewEscalation.....	??
B.2.35. Frontend::Agent::Ticket::ViewForward.....	??
B.2.36. Frontend::Agent::Ticket::ViewFreeText	??
B.2.37. Frontend::Agent::Ticket::ViewHistory	??
B.2.38. Frontend::Agent::Ticket::ViewMailbox.....	??
B.2.39. Frontend::Agent::Ticket::ViewMerge.....	??
B.2.40. Frontend::Agent::Ticket::ViewMove	??
B.2.41. Frontend::Agent::Ticket::ViewNote	??
B.2.42. Frontend::Agent::Ticket::ViewOwner	??
B.2.43. Frontend::Agent::Ticket::ViewPending	??
B.2.44. Frontend::Agent::Ticket::ViewPhoneNew	??
B.2.45. Frontend::Agent::Ticket::ViewPhoneOutbound	??
B.2.46. Frontend::Agent::Ticket::ViewPriority	??
B.2.47. Frontend::Agent::Ticket::ViewQueue.....	??
B.2.48. Frontend::Agent::Ticket::ViewResponsible.....	??
B.2.49. Frontend::Agent::Ticket::ViewSearch	??
B.2.50. Frontend::Agent::Ticket::ViewStatus	??
B.2.51. Frontend::Agent::Ticket::ViewZoom.....	??
B.2.52. Frontend::Agent::TicketOverview	??
B.2.53. Frontend::Agent::ToolBarModule	??
B.2.54. Frontend::Customer	??
B.2.55. Frontend::Customer::ModuleMetaHead	??
B.2.56. Frontend::Customer::ModuleRegistration	??
B.2.57. Frontend::Customer::Preferences	??
B.2.58. Frontend::Customer::Ticket::ViewNew	??
B.2.59. Frontend::Customer::Ticket::ViewSearch.....	??
B.2.60. Frontend::Customer::Ticket::ViewZoom.....	??
B.2.61. Frontend::Queue::Preferences	??
B.2.62. Frontend::SLA::Preferences	??
B.2.63. Frontend::Service::Preferences	??

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

List of Tables

3-1. Needed Perl modules for OTRS	??
3-2. Description of several cron job scripts.	??
5-1. Default groups after OTRS has been installed	??
5-2. Rights in the groups of OTRS	??
5-3. Events for auto answers.....	??
5-4. Function of the different X-OTRS-headers	??
A-1. Mailinglists.....	??

Preface

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

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

Since the book tries to satisfy the needs of OTRS admins or OTRS newbies and since the quality should be as good as possible, we need your feedback. Please write to us if you find something missing in this book, if things are not explained well enough or even if you see spelling mistakes, grammatical errors or typos. Any kind of feedback is highly appreciated and should be made via our bug tracking system on <http://bugs.otrs.org>. 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 specific. A short example shows the advantages of using a ticket system.

1.1. What is a trouble ticket system and why do you need it?

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

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

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

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

Let's imagine that Mr Smith makes a request to Max's company and his message is processed by OTRS.

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

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

1.2. What is a trouble ticket?

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

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

Chapter 2. OTRS - Open Ticket Request System

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

2.1. Basics

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

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

2.2. Features

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

The features of OTRS

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

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

- ASP support (active service providing).
- Linking several objects is possible, e.g. tickets and FAQ entries.
- Integration of external back-ends for the customer data, e.g. via AD, eDirectory or OpenLDAP.
- Setting up an own ticket identifier, e.g. Call#, Ticket# or Request#.
- The integration of your own ticket counter is possible.
- Support of several database systems for the central OTRS back-end, e.g. MySQL, PostgreSQL, Oracle, DB2).
- Framework to create stats.
- utf-8 support for the front- and back-end.
- Authentication for customers via database, LDAP, HTTPAuth or Radius.
- Support of user accounts, user groups and roles.
- Support of different access levels for several systems components or queues.
- Integration of standard answer texts.
- Support of sub queues.
- Different salutations and signatures can be defined for every queue.
- Email notifications for admins.
- Information on updates via mail or the web interface.
- Escalation for tickets.
- Support for different time zones.
- Simple integration of own add-ons or applications with the OTRS API.
- Simple creation of own front-ends, e.g. for X11, console.

2.2.1. Top new features of OTRS 3.0

Context

- User Centered redesign of the Graphical User Interface which results in a dramatic shift from a comprehensive but static to a more powerful and dynamic application using state-of-the art technologies like Ajax, xHTML and optimized CSS.

New Ticket and Article Indicator

- This new feature has been implemented on both ticket and article level. It allows an agent at a glance to check for any updates within a ticket or on the article level to check for new and unread articles. You benefit from increased transparency and decreased response times.

Optimized Fulltext Search

- The new search feature allows you to flexibly customize the way you browse the information base. Options the new search feature provides range from single search-string searches to complex

multi-string boolean search operations including various operators. You benefit from fully customizable searches according to your needs.

New Ticket Zoom View

- The redesign based on Ajax technology allows agents to display complex and linked information structures in real-time while keeping the agents' current working environment. The agent will benefit from increased orientation and increased workflow efficiency.

Global Ticket Overviews

- Well known from OTRS 2.4 the global ticket overviews have been optimized to achieve increased inter- activity. Depending on the use case and preferences of your agents they can easily change the ticket overviews layout according to their special needs. Options are small, medium and large, each providing a different degree of information details.

Accessibility

- The redesign includes common accessibility standards WCAG and WAI-ARIA which also allows disabled users to better interact with OTRS Help Desk. The US Rehabilitation Acts Section 508 has been fulfilled.

New Customer Interface

- The customer web front-end can be integrated to your organizations intranet and is fully integrated into the redesigned help desk system.

Archive Feature

- OTRS 3.0 now offers a new archiving feature. With a separated archive you'll benefit from a reduced time spent for searches and increased display of results.

2.2.2. New features of OTRS 2.4

Licensing changed to AGPL Version 3

- Why AGPL instead of GPL? - AGPL and GPL are identical, with one exception: For software used in an SaaS environment Copyleft is effective in AGPL - which is not the case when using GPL. Keeping in mind the growing world of SaaS, ((otrs)) wants to ensure that future developments continue to return to the OTRS community. This is the reason for the switch to AGPL.

Why v3 instead of v2? - GPL v2 is getting older and has, especially in the USA, various legal uncertainties. In the opinion of ((otrs)) GPL v3 is keeping the spirit of GPL v2, and at the same time has been tailored to new needs. ((otrs)) views GPLv3, more specifically AGPLv3, as being the best balanced Copyleft Open Source License available today, offering Protection for copyright owners and users and providing the best security under the law.

New Management Dashboard

- The need for a system-spanning, next to real-time, and personalized presentation of useful information led to an integrated Management Dashboard. It is possible to create plug-ins to display content from individual extensions alongside the standard content. Standard plug-ins are:
- Ticket volume (new & open) from the last 24h, 48h and 72h
- Calendar including an overview of upcoming events (escalations, auto-unlocks, etc.)
- System-wide overview of ticket distribution within the queues
- First Response Time/Solution Time of Queues
- Integration of RSS

New Standard Reports

- The new reports provided with OTRS 2.4 are:
- Created Tickets
- Closed Tickets
- SLA Analysis
- Required working time per customer / per queue
- Solution time analysis per customer / per queue
- Answer time analysis per customer / per queue

New Master/Slave Ticket Feature

- With the Master/Slave Ticket, it is possible to link multiple tickets of a similar nature, and handle them collectively. As soon as the problem is solved, only the master ticket must be closed. All other tickets will be closed automatically, and the solution text for the master ticket will be sent to all customers of slave tickets.

A new link type 'Slave' will be available. All tickets with this Type of link will inherit the following actions from their Master ticket:

- Status change
- Email answers
- Change in FreeText fields
- Notes
- Pending time changes
- Priority changes
- Owner changes
- Responsibility changes

New Rich-Text/HTML E-Mail Support (WYSIWYG)

- With this feature, it is now possible to write e-mails, notes, and notifications in rich text format (HTML format). Using a WYSIWYG editor (What You See Is What You Get), it is possible to comfortably write using formatted text and even include in-line pictures.

New Out-Of-Office Feature

- With this new feature it is possible for all users to activate "out-of-office" to notify colleagues and OTRS of the period of their absence. The out-of-office feature is active for a time frame set by the user. Activation of this feature has the following effects:

In the lists in which an agent can be selected as owner or responsible (i.e. Ticket creation or changing ownership), the period of absence and the time till return will be shown behind the user's name. This will help making the absence of the user more transparent.

If an agent receives a follow-up during a period of absence, the ticket is automatically unlocked and a notification is sent to all agents in the queue. This allows immediate reaction to the customer follow-up by another service employee.

New Ticket Overviews and global Bulk Action

- Flexibility of presentation within the ticket overview is a must. Based on the "S/M/L" (Small/Medium/Large) Ticket View every agent has the possibility to change the view for each type of overview (Queue View, Status View, etc) on-the-fly with a simple mouse click on the appropriate icon. This allows for the highest possible level of individualization and adjustment to any operational situation.

Additionally, decentralization of the Bulk Action feature integrated the Bulk Action in all ticket overviews (Bulk Action allows processing of multiple tickets at a time).

Postmaster Filter recognizes Follow-Ups to internal forwarded messages

- Currently, e-mail replies to forwarded articles arrive in OTRS as email-external. The problem is that the answers to these forwarded articles can be seen by the customer in the web- interface. Although it is possible to classify e-mails of an entire domain as email-internal, this only shifts the problem. Also, such step makes it impossible to properly service customers in the domain, as the customer would not be able to track tickets in the customer web-interface any more. With this new feature, e-mail replies can be traced back, and email- internal or email-external will be set based upon the original Forward-Article type.

Configurable event based notifications

- Until now, a very inflexible notification could be sent to an agents and customers, for example Agent: New Ticket or Customer: Status Change. In order to make the notification system more flexible, a complete overhaul was performed on the messaging mechanism. The new system allows messaging to agents, customers, or a dedicated email address, based on the event taking place.

With this, it is now possible to just inform the customer when the ticket has been closed. Or, for example, when a VIP customer creates a ticket, a message can be sent to a specific address. Events (i.e. TicketCreate, TicketStateUpdate, TicketPriorityUpdate, ArticleCreate), and all known message variables (i.e. <OTRS_TICKET_TicketNumber> <OTRS_TICKET_Priority>), are freely selectable for creating triggered messages via the web interface.

READ-ONLY Permissions and Notifications with watched Tickets

- In the current release of OTRS it is possible for a user to maintain a Watched Tickets List. This feature is dealing with tickets marked as "subscribed" by a user. It has the advantage that users no longer lose track of tickets marked as "subscribed", and are able to view them on an individual list. The "Read-Only" Feature - Up to now, tickets marked as "subscribed" were shown in a list, however, the agent could only actually view them if they were in a queue for which the agent had read permissions. With the "Read-Only" Feature, agents subscribed to a ticket always have read permissions on the ticket, even if the ticket is moved to a queue where the agent has no permissions. "Notify" Feature - Via a personalized setting, every agent can define whether or not to receive notifications about tickets, just as the owner and responsible of a ticket would receive. This allows for active tracking of watched tickets.

Secure SMTP

- OTRS can receive and send mails in multiple ways. All currently available methods for receiving emails have been implemented within OTRS 2.3 (POP3, POP3S, IMAP, IMAPS). Until now, there were two options for sending emails: using a local MTA (Sendmail, Postfix, etc.) or per SMTP. In OTRS 2.4.x, SMTPS (Secure SMTP) has been implemented in order to keep up to the growing security standards.

2.2.3. New features of OTRS 2.3

Performance

- Data base- and code-improvements increase lead to a general performance gain of up to 20%.
- The support of an indexed full text search has been added. The feature is disabled per default because additional disc space is needed. The expected performance gain is 50%.
- Reduced reloads by using AJAX technology.
- Instead of an ongoing recalculating of the escalation time during run time, it is only recalculated when it changes due to an event in OTRS. It is then being stored in the ticket object which allows a direct access of external reporting tools to the data base as well as a more efficient reporting on escalations. This will also lead into a substantial performance improvement.

Search functionality

- Support of logical expressions: ticket-, customer- and FAQ- search supports logical expressions, utilizing the AND, OR and ! operators as well as structuring expressions with parentheses.
- Search for ticket numbers by using the Browser OpenSearch feature (OpenSearch format).
- Search for ticket titles in the agent ticket search form and in the generic agent.
- Search for ticket close time in the agent ticket search form and in the generic agent.

Ticket zoom and ticket move

- Expand/Collapse of articles: the article view can be expanded to display all articles at once. The current article will remain in focus, and the preceding, or following articles will be displayed.
- Structured article tree - The article tree has been changed to a table.
- Printing of articles has been realized.
- The ticket title of linked tickets are displayed in case of a mouse over action.
- Merged tickets are displayed crossed out.
- Multiple files can be attached while moving a ticket using the ticket move mask.

Ticket FreeText and FreeTime opportunities

- When splitting a ticket, all FreeText and Free Time data will be copied to the new ticket.
- Ticket Free Time fields can be declared as mandatory.
- A URL can be configured that takes the value of a FreeText field and displays it as an URL link in the ticket.
- Added X-OTRS-TicketTime and X-OTRS-FollowUp-TicketTime email headers.

IMAP, IMAPS and POP3S support

- With OTRS 2.3 additionally to POP3, POP3S, IMAP and IMAPS is supported to fetch mails from your MTA.

Security

- In case of a lost password, OTRS is sending an e-mail to the user with a "password reset link". After clicking this link the new password is sent to the user in a second e-mail.

Notifications and escalations

- All agents that have a read permission on a certain queue can be selected for notification.
- An escalation view has been added that displays all tickets sorted by their remaining time to escalation.

2.2.4. New features of OTRS 2.2

New features of OTRS 2.2

- Support of Services and SLAs: As major step towards IT Service Management OTRS 2.2 brings in the new attributes 'Service' and 'Service Level Agreements (SLA)'. While creating a new ticket a service requester has to select both a service (e. g. email-service) and a related SLA. SLA attributes are "response time", "update time" and "solution time". These attributes are used by the IT Service Organization for notification and escalation purposes in order to keep the agreed SLA. Service- and SLA-related informations within incoming email headers might also be used by the Postmaster-Filter as already known before.
- Support of native ticket types: Ticket types can now be managed over the admin interface. You do not longer need to use ticket free text field for this purpose. Installations which already use a ticket free text field for ticket type classification do not need to migrate. This feature will also be shown in zoom and print view for agents and customers and can be changed via the agent interface.

- Support of multiple authentication backends: Added support of multi authentication feature for agent and customer backend. By using this feature it is now possible to use multiple trusted authentication sources (i.e. use LDAP as a first source for authentication and in case of an unsuccessful authentication try a SQL DB as a second source).
- Support of different password crypt types: Supported and permitted password authentication types for agent and/or customer login are:
 - `unix_crypt()`
 - `md5()`
 - `plain()`

Default is `'unix_crypt()'`. The administrator can easily switch from `unix_crypt()` to `md5()` on the fly by changing the related configuration parameter.

- Changed OTRS internal CustomerUser structure: Customer datas (CustomerUser) are restructured and split into the objects "CustomerCompany" and "CustomerUser" within the internal OTRS database. Company related attributes like company name and address will be managed separately from those attributes related to a single contact (e.g. first name, surname, phone etc.).

This information will be shown in zoom and print like the current customer info.

- Enhanced OPM-Format: OPM Packages are now delivered with an additional on-line help. The introduction page will inform the user about new features and will also give a short overview about the next steps to do after having installed the packages.
- PostMaster-Admin-Interface: The improvement has lead to a more detailed definition of those email addresses which have to be matched by PostMaster filter.
- SysConfig Improvement: A quicker page view and page saving leads to essential time savings during configuration process.

2.2.5. New features of OTRS 2.1

New features of OTRS 2.1

- Stable Support of Microsoft SQL Server: starting this version OTRS offers an interface to MS SQL Server databases. OTRS supports all popular operating systems like Linux, Windows, UNIX, Mac OS X etc. as well as all major databases like MS SQL, MySQL, PostgreSQL, Oracle and DB2.
- Multi-Calendar Function: This Feature simplifies working in distributed surroundings. National teams are supported by permitting different local working schedules and public holiday regulations. International organizations are additionally supported by a time-zone feature.
- New Statistics-Framework: Gain access and define your Reports- and Statistics in a more flexible way than ever. A powerful and easy to use Wizard will guide you through the process. Once defined OTRS may deliver periodically needed Reports automatically to a predefined Mailing List. Export- and

Import functions allow to transfer Statistics between different OTRS Installations. So you can define and test your Statistics before loading them up to your production System.

- **Optimized Synchronization of authorizations from LDAP-Directory Services:** That simplifies the connection of all known LDAP-Directory Services like Microsoft Active Directory, OpenLDAP, Novell eDirectory, Oracle Internet Directory, etc. A useful Feature which saves time and effort of maintaining authorizations in the local Database of OTRS.
- **PDF-Generator:** OTRS now supports the Output of Reports and Statistics as well as the results of a ticket search and ticket zoom in the PDF-Format.
- **New implemented Knowledge Database / FAQ-Module:** OTRS makes allowance for the need of an efficient Knowledge Management within daily Service Request handling. Fundamental in this context is a quick access towards existing problem-solving Know-How and standardized answers. Advantages: optimized search-functions, more effective administration of FAQ-Articles, faster Navigation and opportunities to rate an existing FAQ-Article.
- **Tracking of Sub-orders and Support of hierarchical Team-Structures:** New Features like the Ticket Supervisor and the Ticket Watchers support working within hierarchical Teams and enable you to define and depute Work packages temporarily to another Service Agent. While the Ticket Supervisor enables authorized Team members to depute a Ticket to another Service Agent without losing the right to edit this ticket the Ticket Watcher allows you to track a ticket passive on the basis of an individual watch list.
- **Optimized Follow-Up-Detection:** In addition to the subject heading, OTRS also checks for referable ticket numbers inside the Email-body and further attachments. This new feature aims to automatically assign an incoming Service Request to an existing Incident much faster than before.

2.3. Hardware and software requirements

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

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

For the web server, we recommend using apache 2, because its module `mod_perl` improves greatly the performance of OTRS. If you can't use apache, OTRS should run on any web server that can execute Perl scripts.

For database back-ends, you can use MySQL, PostgreSQL, Oracle, MSSQL or DB2. If you use MySQL you have the advantage that the database and some system settings can be configured during the

installation, through a web front-end.

For Perl, we recommend using at least version 5.8.8. 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).

Software requirements

Perl

- Perl 5.8.8 or higher

Webserver

- Apache2 + mod_perl2 or higher (recommended, mod_perl is really fast!)
- Webserver with CGI support (CGI is not recommended)
- IIS 6 or higher

Databases

- MySQL 4.1 or higher
- PostgreSQL 8.0 or higher
- Oracle 10g or higher
- DB2 8 or higher
- MSSQL 2000 or higher

The section in the manual about installation of Perl modules describes in more detail how you can set up those which are needed 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 dependencies of the Perl modules needed.

2.4. Community

OTRS has a large user community. Users and developers discuss about OTRS and interchange information on related issues through the mailing-lists . There is available help for questions regarding the installation, configuration, usage, localization and development. Software bugs can be reported on the bug tracking system , so that they reach the responsible developers directly and without getting lost.

The homepage of the OTRS community is: <http://www.otrs.org> (<http://www.otrs.org>) (see Figure 2.1 below).

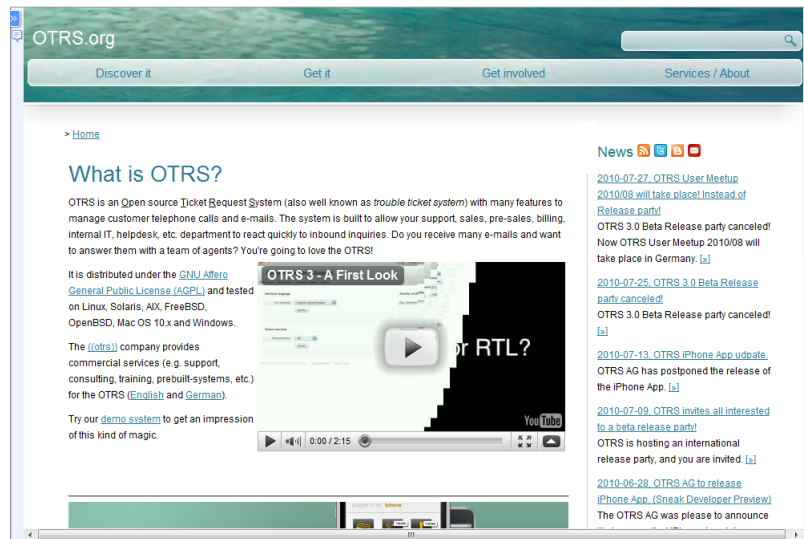


Figure 2.1. OTRS community's homepage.

2.5. Commercial Support for OTRS

Commercial support for OTRS is also available. You can find the available options on the website of OTRS AG, the company behind OTRS: <http://www.otrs.com/en/support> (<http://www.otrs.com/en/support>) .

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

You can find more detailed information about OTRS AG on <http://www.otrs.com> (<http://www.otrs.com>) and have contact via email on sales@otrs.com (<mailto:sales@otrs.com>) .

Chapter 3. Installation / Upgrade 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 from source or with a binary package, for example an RPM or with the Windows Installer.

The configuration of the web and database servers, the interface between OTRS and the database, the installation of additional Perl modules, setting proper access rights for OTRS, setting up the cron jobs for OTRS and some basic settings in the configuration files of OTRS are covered in this chapter.

When you follow the steps in this chapter, you will have a running OTRS system installed on your computer, where you can login and administrate the system via the web interface.

3.1. The simple way - Installation of pre-built packages

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

3.1.1. Installing the rpm on a SUSE Linux server

This section describes the installation of a pre-built rpm package on a SUSE Linux distro. We tested all recent SLES and OpenSUSE versions. Before you start the installation please have a look on <http://www.otrs.org/downloads> (<http://www.otrs.org/downloads>) 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 it solves the package dependencies automatically.

If you decide to install OTRS via the command line and rpm, first you have to manually install the needed Perl modules. Assuming you saved the file `otrs.rpm` into the directory `/tmp`, you can execute the command specified in the Script 3.1 to install OTRS.

Note: If you install it on SLES9 or OES, follow this description to install the missing perl-GD packages.
<http://faq.otrs.org/otrs/public.pl?FAQID=49>

```
linux:~ # rpm -i /tmp/otrs-xxx.rpm
otrs #####
Check OTRS user (/etc/passwd)... otrs exists.

Next steps:

[SuSEconfig]
Execute 'SuSEconfig' to configure the web server.

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

[install the OTRS database]
Use a web browser 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:~ #
```

Script 3.1. Command to install OTRS.

After the installation of the OTRS rpm package, you have to run SuSEconfig, as shown in the Script 3.2.

```
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:~ #
```

Script 3.2. Running the SuSEconfig command.

The installation of the OTRS rpm is finished. Restart your web server to load the OTRS specific changes in its configuration, as shown in the Script 3.3.

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

Script 3.3. Restarting the web server.

In the next step you have to setup the OTRS database, as described in the section 3.2.4.

3.1.2. Installing OTRS on a CentOS system

On the OTRS Wiki you can find detailed instructions for setting up OTRS on a CentOS system. Please note that these instructions will also apply to RedHat systems as those use the same source:

http://wiki.otrs.org/index.php?title=Installation_of_OTRS_3.0b1_on_CentOS_5.5
(http://wiki.otrs.org/index.php?title=Installation_of_OTRS_3.0b1_on_CentOS_5.5) .

3.1.3. Installing OTRS on a Debian system

On the OTRS Wiki you can find detailed instructions for setting up OTRS on a Debian system:

http://wiki.otrs.org/index.php?title=Installation_on_Debian_5.04_lenny
(http://wiki.otrs.org/index.php?title=Installation_on_Debian_5.04_lenny) .

3.1.4. Installing OTRS on a Ubuntu system

On the OTRS Wiki you can find detailed instructions for setting up OTRS on an Ubuntu system:

[http://wiki.otrs.org/index.php?title=Installation_on_Ubuntu_Lucid_Lynx_\(10.4\)](http://wiki.otrs.org/index.php?title=Installation_on_Ubuntu_Lucid_Lynx_(10.4))

([http://wiki.otrs.org/index.php?title=Installation_on_Ubuntu_Lucid_Lynx_\(10.4\)](http://wiki.otrs.org/index.php?title=Installation_on_Ubuntu_Lucid_Lynx_(10.4))) .

3.1.5. Installing OTRS on Microsoft Windows systems

Installing OTRS on a Microsoft Windows system is very easy. Download the latest installer for Win32 from <http://www.otrs.org/downloads/> (<http://www.otrs.org/downloads/>) 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 that you will be able to login as OTRS administrator and configure the system according to your needs. To log in as OTRS administrator use the username root@localhost and the default root password.

Warning

Please change the password for the root@localhost account as soon as possible.

Important: The Win32 installer for OTRS contains all needed components for OTRS, i.e. 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 want to install OTRS from source, first download the source archive as .tar.gz, .tar.bz2, or .zip file from <http://www.otrs.org/downloads/> (<http://www.otrs.org/downloads/>)

Unpack the archive for example with **tar** into the directory /opt and rename the directory from otrs-3.0 to otrs (see Script 3.4 below).

```
linux:/opt# tar xf /tmp/otrs-3.0.tar.gz
linux:/opt# mv otrs-3.0 otrs
linux:/opt# ls
```

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

Script 3.4. Firsts steps to install OTRS.

Because the modules of OTRS should not be executed with root rights, you should add a new user for OTRS as the next step. The home directory of this new user should be `/opt/otrs`. If your web server is not running with the same user rights as the new otrs user, which is the case on most systems, you have to add the new otrs user to the group of the web server user (see Script 3.5 below).

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

Script 3.5. Adding a new user and adding it to a group.

Now you have to copy some sample configuration files. The system will later use the copied files. The files are located in `/opt/otrs/Kernel` and `/opt/otrs/Kernel/Config` and have the suffix `.dist` (see Script 3.6 below).

```
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# cp GenericAgent.pm.dist GenericAgent.pm
```

Script 3.6. Copying some sample files.

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

```
otrs.SetPermissions.pl { Home directory of the OTRS user } { --otrs-user= OTRS user } {
--web-user= Web server user } [ --otrs-group= Group of the OTRS user ] [ --web-group=
Group of the web server user ]
```

If your web server is running with the same user rights as OTRS, the command to set the proper access rights is **otrs.SetPermissions.pl /opt/otrs --otrs-user=otrs --web-user=otrs**. On SUSE systems the web server is running with the user rights of `wwwrun`. On Debian-based systems this is `www-data`. You would use the command **otrs.SetPermissions.pl /opt/otrs --otrs-user=otrs --web-user=wwwrun --otrs-group=nogroup --web-group=www** to set the proper access rights.

3.2.2. Installation of Perl modules

OTRS needs some additional Perl modules, as described in the Table 3-1. If you install OTRS from source, you'll have to install these modules manually. This can either be done with the package manager of your Linux distribution (`yast`, `apt-get`) or, like described in this section, through the Perl shell and CPAN. If you're using ActiveState Perl, for instance on Windows, you could use PPM, the built-in Perl Package Manager. If possible we advise you to use your package manager.

Table 3-1. Needed Perl modules for OTRS

Name	Description
DBI	Establishes a connection to the database back-end.
DBD::mysql	Contains special functions to connect to the MySQL database back-end (only required if MySQL is used).
DBD::pg	Contains special functions to connect to the PostgreSQL database back-end (only required if PostgreSQL is used).
Digest::MD5	Allows the use of the md5 algorithm.
CSS::Minifier	Minifies a CSS file and writes the output directly to another file.
Crypt::PasswdMD5	Provides interoperable MD5-based crypt functions.
MIME::Base64	Encodes / decodes base64 strings, e.g. for mail attachments.
JavaScript::Minifier	Minifies a JavaScript file and writes the output directly to another file.
Net::DNS	Perl interface to the domain name system.
LWP::UserAgent	Processes HTTP requests.
Net::LDAP	Perl interface to a LDAP directory (only required a LDAP back-end is used).
GD	Interface to the GD graphics library (only required if the OTRS stats module is used).

Name	Description
GD::Text, GD::Graph, GD::Graph::lines, GD::Text::Align	Some more text- and graphic tools for the GD graphics library (only required if the OTRS stats module is used).
PDF::API2, Compress::Zlib	Needed to generate the PDF output for reports, search results and for the ticket print view.

You can verify which modules you need to install with **otrs.CheckModules.pl**. This script is located in the `bin` directory, in the home directory of the OTRS user (see Script 3.7 below).

Please note that some modules are optional.

```
linux:~# cd /opt/otrs/bin/
linux:/opt/otrs/bin# ./otrs.CheckModules.pl
  o CGI.....ok (v3.49)
  o Crypt::PasswdMD5.....ok (v1.3)
  o CSS::Minifier.....ok (v0.01)
  o Date::Format.....ok (v2.24)
  o Date::Pcalc.....ok (v1.2)
  o DBI.....ok (v1.609)
  o DBD::mysql.....ok (v4.013)
  o Digest::MD5.....ok (v2.36_01)
  o Encode::HanExtra.....ok (v0.23)
  o GD.....ok (v2.44)
    o GD::Text.....ok (v0.86)
    o GD::Graph.....ok (v1.44)
    o GD::Graph::lines.....ok (v1.15)
    o GD::Text::Align.....ok (v1.18)
  o IO::Scalar.....ok (v2.110)
  o IO::Wrap.....ok (v2.110)
  o JavaScript::Minifier.....ok (v1.05)
  o JSON.....ok (v2.21)
    o JSON::PP.....ok (v2.27003)
    o JSON::XS.....Not installed! (Optional - Install it for faster AJAX/J
  o LWP::UserAgent.....ok (v5.829)
  o Mail::Internet.....ok (v2.06)
  o Mail::POP3Client.....ok (v2.18 )
    o IO::Socket::SSL.....ok (v1.31)
  o MIME::Base64.....ok (v3.07_01)
  o MIME::Tools.....ok (v5.428)
  o Net::DNS.....ok (v0.65)
  o Net::POP3.....ok (v2.29)
  o Net::IMAP::Simple.....ok (v1.1916)
    o Net::IMAP::Simple::SSL.....ok (v1.3)
  o Net::SMTP.....ok (v2.31)
    o Authen::SASL.....ok (v2.15)
```



```

    o Net::SMTP::SSL.....ok (v1.01)
  o Net::LDAP.....ok (v0.4001)
  o PDF::API2.....ok (v0.73)
    o Compress::Zlib.....ok (v2.008)
  o SOAP::Lite.....ok (v0.712)
  o Text::CSV.....ok (v1.18)
    o Text::CSV_PP.....ok (v1.26)
    o Text::CSV_XS.....Not installed! (Optional - Optional, install it for fas
  o XML::Parser.....ok (v2.36)
linux:/opt/otrs/bin#

```

Script 3.7. Checking needed modules.

You should strive to install the missing modules from your linux distributions package management system. In that way, the packages will be automatically updated when new versions are available or when security issues are found. Please refer to your distributions documentation on how to install additional packages. If the (correct version of) the module is not available from the package repositories, you can also install from CPAN, the Comprehensive Perl Archive Network.

To install one of the modules from above via 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 will let you know, if other modules are needed.

Execute also the two commands **perl -cw bin/cgi-bin/index.pl** and **perl -cw bin/otrs.PostMaster.pl** after changing into the directory `/opt/otrs`. If the output of both commands is "syntax OK", your Perl is properly set up (see Script 3.8 below).

```

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

```

Script 3.8. Checking syntax.

3.2.3. Configuring the apache web server

This section describes the basic configuration of the apache web server with `mod_cgi` for OTRS. The web server should be able to execute CGI scripts. OTRS won't work if the Perl scripts cannot be parsed. Check the configuration 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 already 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, change to the `conf.d` directory and copy the appropriate template in `/opt/otrs/scripts/apache2-httpd.include.conf` to a file called `otrs.conf` in the Apache configuration directory.

Restart your web server to load the new configuration settings. On most systems you can start/restart your web server with the command `/etc/init.d/apache2 restart`(see Script 3.11 below).

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

Script 3.11. Restarting the web server.

Now your web server should be configured for OTRS.

If you choose to increase performance, and you can install `mod_perl`, then you can leave `mod_cgi` off, and configure the apache web server for use with `mod_perl`, in the following manner.

Please ensure that `mod_perl` is installed and loaded, in order to take advantage of this feature. Due to the nature of the start-up script, your server will not fail to start, if the `mod_perl` is not properly loaded or compiled in your apache web server, unless `mod_cgi` is also on. Technically speaking you can leave `mod_cgi` on as well, but you should not.

Search your `/etc/apache*` directory for `mod_perl.so` (see Script 3.12 below) to see if the module is already loaded somewhere, or not.

```
#:/ grep -Rn mod_perl.so /etc/apache*
```

Script 3.12. Searching for `mod_perl`.

When you use the appropriate start script (listed above), and the module is loaded, the script (when commented in) `/opt/otrs/scripts/apache2-perl-startup.pl` can be used to load the perl modules into memory one time, saving on load times and increasing performance.

3.2.4. Configuring the database

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

If you use MySQL as the database back-end, you can use the OTRS web installer:

`http://localhost/otrs/installer.pl` (`http://localhost/otrs/installer.pl`)

When the web installer starts, please follow the next steps to setup your system:

1. Check out the information about the OTRS offices and click on next to continue (see Figure 3.1 below).

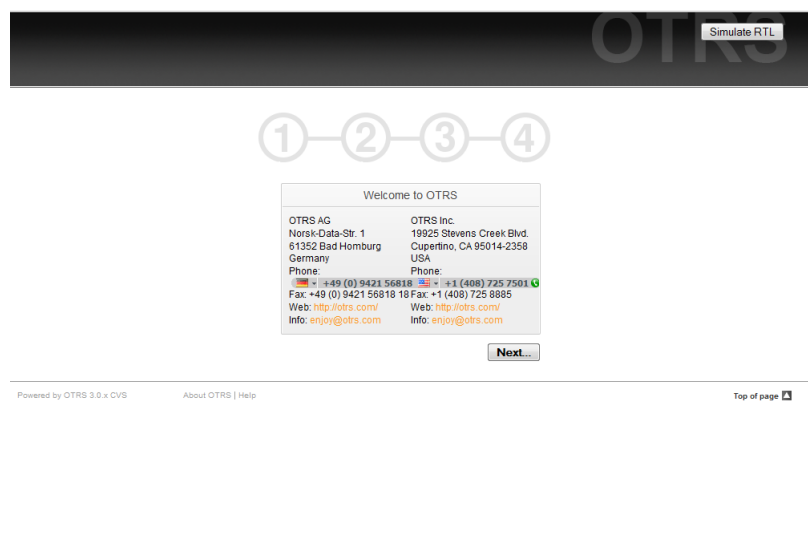


Figure 3.1. Welcome screen.

2. Read the GNU Affero General Public License (see Figure 3.2 below) and accept it, by clicking the corresponding button at the bottom of the page.



Figure 3.2. GNU Affero General Public License.

3. Provide the username and password of the administrator, the DNS name of the computer which hosts OTRS and the type of database system to be used. After that, check the settings (see Figure 3.3 below).

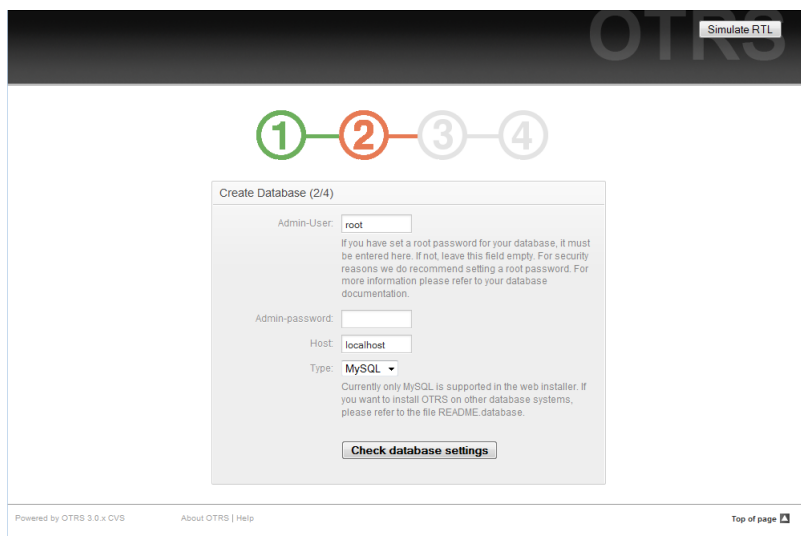


Figure 3.3. Database initial settings.

If the checking was successful, you will get a notification. Press OK to continue (see Figure 3.4 below).

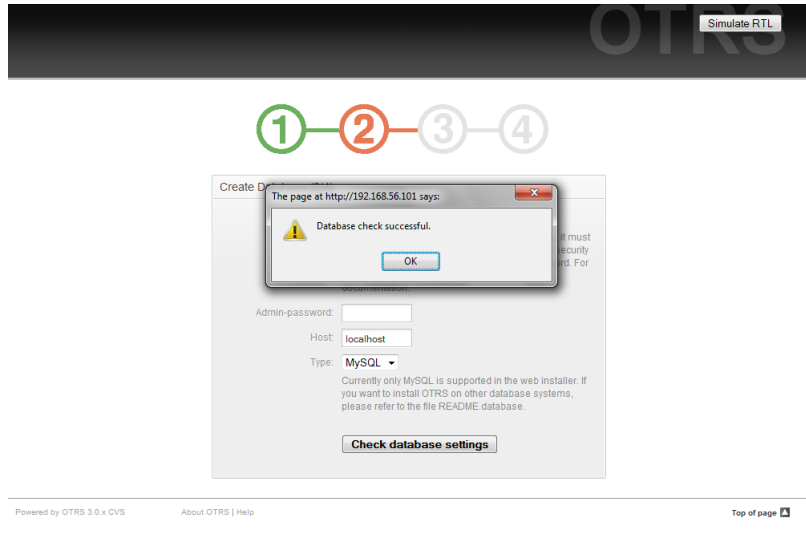


Figure 3.4. Successful checking notification.

4. Create a new database user, choose a name for the database and click on next (see Figure 3.5 below).

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

Create Database (2/4)

Admin-User:

If you have set a root password for your database, it must be entered here. If not, leave this field empty. For security reasons we do recommend setting a root password. For more information please refer to your database documentation.

Admin-password:

Host:

Type:

Currently only MySQL is supported in the web installer. If you want to install OTRS on other database systems, please refer to the file README.database.

Database-User (New)

User:

A new database user with limited rights will be created for this OTRS system.

Password:

default: 'otrs'

DB connect host:

Database

Name:

Action: ☒ Create ☐ Delete

Next...

Figure 3.5. Database settings.

If the database and its user were successfully created, you will get a setup notification, as shown in the Figure 3.6. Click next to go to the next screen.

OTRS Simulate RTL

1 — 2 — 3 — 4

Create Database (2/4)

Creating database 'otrs': **Done.**

Creating tables 'otrs-schema.mysql.sql': **Done.**

Inserting initial inserts 'otrs-initial_insert.mysql.sql': **Done.**

Foreign Keys 'otrs-schema-post.mysql.sql': **Done.**

Creating database user 'otrs@localhost': **Done.**

Reloading grant tables: **Done.**

====> Database setup successful!

Next...

Powered by OTRS 3.0.x CVS About OTRS | Help Top of page

Figure 3.6. Successfully database setup notification.

5. Provide all the required system settings and click next (see Figure 3.7 below).

The screenshot shows the 'System Settings (3/4)' configuration window in the OTRS framework. At the top, there is a progress bar with four numbered circles: 1 (green), 2 (green), 3 (red), and 4 (grey). The 'System Settings' form includes the following fields:

- SystemID:** A dropdown menu set to '10'. Below it, a note states: 'The identity of the system. Each ticket number and each http session id starts with this number.'
- System FQDN:** A text input field containing 'yourhost.example.c'. Below it, a note states: 'Full qualified domain name of your system.'
- AdminEmail:** A text input field containing 'admin@example.c'. Below it, a note states: 'Email of the system admin.'
- Organization:** A text input field containing 'Example Company'.
- Log:**
 - LogModule:** A dropdown menu set to 'File'. Below it, a note states: 'Used log backend.'
 - LogFile:** A text input field containing '/tmp/otrs.log'. Below it, a note states: 'Logfile just needed for File-LogModule!'
- Webfrontend:**
 - Default language:** A dropdown menu set to 'English (United States)'. Below it, a note states: 'Used default language.'

Figure 3.7. System settings.

6. If you want, you can provide the needed data to configure your inbound and outbound mail or skip this step by pressing the right button at the bottom of the screen (see Figure 3.8 below).

The screenshot shows the 'Mail configuration (3/4)' configuration window in the OTRS framework. At the top, there is a progress bar with four numbered circles: 1 (green), 2 (green), 3 (red), and 4 (grey). The 'Mail configuration' form includes the following sections:

- Configure Outbound Mail:**
 - Outbound mail type:** A dropdown menu set to 'SMTP'. Below it, a note states: 'Select outbound mail type.'
 - SMTP host:** A text input field.
 - SMTP authentication:** A checkbox. Below it, a note states: 'Does your SMTP host need authentication?'
- Configure Inbound Mail:**
 - Inbound mail type:** A dropdown menu set to 'IMAP'. Below it, a note states: 'Select inbound mail type.'
 - Inbound mail host:** A text input field.
 - Inbound mail user:** A text input field. Below it, a note states: 'User for inbound mail.'
 - Inbound mail password:** A text input field.

Figure 3.8. Mail configuration.

7. Restart the OTRS service now, to use the new configuration settings shown in the Script 3.13.

```
linux:~ # rcotrs restart-force
Shutting down OTRS
  Disable /opt/otrs/bin/otrs.PostMaster.pl ... done.
no crontab for otrs
  Shutting down cronjobs ... failed!
Shutting down OTRS (completely)
  Shutting down Apache ... done.
  Shutting down MySQL ... 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/otrs.PostMaster.pl ... done.
  Checking otrs spool dir... done.
  Creating cronjobs (source /opt/otrs/var/cron/*) ... done.

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

linux:~ #
```

Script 3.13. Restarting the OTRS service.

Congratulations! Now the installation of OTRS is finished and you should be able to work with the system (see Figure 3.9 below). 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 administrator, using the username `root@localhost` and the password `root`. After that you can configure the system for your needs.

Warning

Please change the password for the `root@localhost` account as soon as possible.

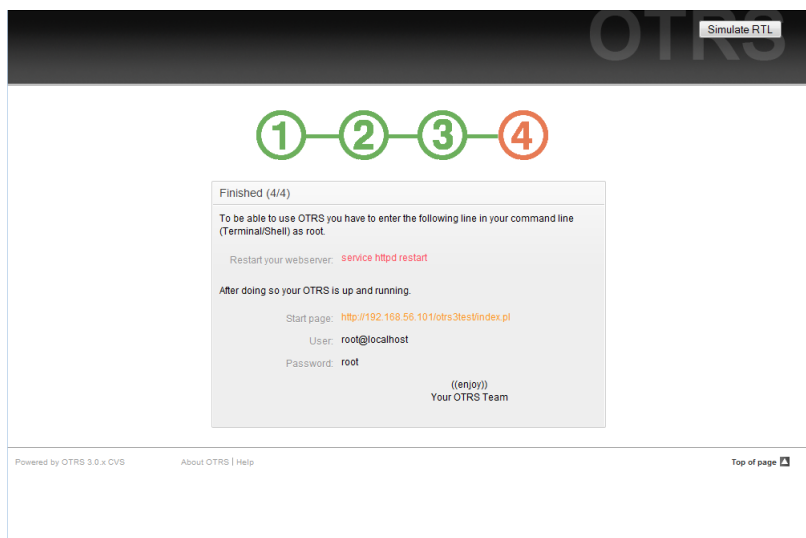


Figure 3.9. Final steps to install OTRS.

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 (see Script 3.14 below).

```
linux:~# cd /opt/otrs/scripts/database/
linux:/opt/otrs/scripts/database# ls
otrs-initial_insert.db2.sql      otrs-schema.mysql.sql
otrs-schema.oracle.sql
otrs-initial_insert.mssql.sql    otrs-schema-post.db2.sql
otrs-initial_insert.mysql.sql    otrs-schema.postgresql.sql
otrs-initial_insert.oracle.sql
otrs-initial_insert.postgresql.sql otrs-schema-post.mssql.sql
otrs-initial_insert.xml          otrs-schema-post.mysql.sql
otrs-schema.db2.sql             otrs-schema-post.oracle.sql
  otrs-schema-post.postgresql.sql
otrs-schema.mssql.sql           otrs-schema.xml
linux:/opt/otrs/scripts/database#
```

Script 3.14. Files needed to create and configure the 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 one of the files `otrs-initial_insert.mysql.sql`, `otrs-initial_insert.db2.sql`, `otrs-initial_insert.oracle.sql`, `otrs-initial_insert.postgresql.sql` or `otrs-initial_insert.mssql.sql`.
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` file to create these (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` located in the home directory of the OTRS user and change the parameters shown in the Script 3.15 to your needs.

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

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

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

# DatabasePw
# (The password of database user.)
```

```
$Self->{DatabasePw} = 'some-pass';
```

Script 3.15. Parameters to be customized.

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 (see Script 3.16 below).

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

Script 3.16. Files needed to create the cron jobs.

All scripts are ending in `.dist`. You should copy them to files with no ending. If you are using bash, you might want to use the command listed in Script 3.17 below.

```
linux:/opt/otrs/var/cron# for foo in *.dist; do cp $foo
'basename $foo .dist'; done
linux:/opt/otrs/var/cron# ls
aaa_base          generic_agent-database.dist  rebuild_ticket_index
aaa_base.dist     generic_agent.dist          rebuild_ticket_index.dist
cache            pending_jobs                session
cache.dist       pending_jobs.dist          session.dist
fetchmail        postmaster                 unlock
fetchmail.dist   postmaster.dist            unlock.dist
generic_agent     postmaster_mailbox
generic_agent-database postmaster_mailbox.dist
linux:/opt/otrs/var/cron#
```

Script 3.17. Copying and renaming all the files needed to create the cron jobs.

Table 3-2 describes the different cron jobs and what they do.

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

Script	Function
aaa_base	Sets the basics for the crontab of the OTRS user.
cache	Removes expired cache entries from disk. Clears the loader cache for CSS and JavaScript files.
fetchmail	Used only if new mails will be fetched with fetchmail into the ticket system.
generic_agent	Executes the jobs of the GenericAgent that are not stored in the database but in own config files.
generic_agent-database	Executes the jobs of the GenericAgent that are stored in the database.
pending_jobs	Checks in the system for pending tickets, and closes them or sends reminders if needed.
postmaster	Checks the message queue of the ticket system and delivers messages that are still in the queues.
postmaster_mailbox	Fetches the mails from the POP3 accounts that were specified in the admin area, in the section for "PostMaster Mail Accounts".
rebuild_ticket_index	Rebuilds the ticket index, which improves the speed of the QueueView.
session	Removes old and no longer needed session ID's.
unlock	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. When this script is executed, it needs a parameter to tell if you want 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 commands specified in the Script 3.18 below to install the cron jobs.

Warning

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

```
linux:/opt/otrs/var/cron# cd /opt/otrs/bin/
linux:/opt/otrs/bin# su otrs
linux:~/bin$ ./Cron.sh start
/opt/otrs/bin
Cron.sh - start/stop OTRS cronjobs
Copyright (C) 2001-2009 OTRS AG, http://otrs.org/
(using /opt/otrs) done
linux:~/bin$ exit
exit
linux:/opt/otrs/bin#
```

Script 3.18. Installing the cron jobs.

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 (see Script 3.19 below).

```
linux:/opt/otrs/bin# crontab -l -u otrs
# --
# cron/aaa_base - base crontab package
# Copyright (C) 2001-2011 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# Who gets the cron emails?
MAILTO="root@localhost"

# --
# cron/cache - delete expired cache
# Copyright (C) 2001-2011 OTRS AG, http://otrs.org/
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# delete expired cache weekly (Sunday mornings)
20 0 * * 0 $HOME/bin/otrs.CacheDelete.pl --expired >> /dev/null
30 0 * * 0 $HOME/bin/otrs.LoaderCache.pl -o delete >> /dev/null
```

```
# --
# cron/fetchmail - fetchmail cron of the OTRS
# Copyright (C) 2001-2011 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# fetch every 5 minutes emails via fetchmail
*/5 * * * * /usr/bin/fetchmail -a >> /dev/null

# --
# cron/generic_agent - otrs.GenericAgent.pl cron of the OTRS
# Copyright (C) 2001-2011 OTRS AG, http://otrs.org/
# --
# --
# 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/otrs.GenericAgent.pl -c "Kernel::Config::GenericAgentMove" >> /dev/nu
# --
# cron/generic_agent - GenericAgent.pl cron of the OTRS
# Copyright (C) 2001-2011 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# start generic agent every 10 minutes
*/10 * * * * $HOME/bin/otrs.GenericAgent.pl -c db >> /dev/null
# --
# cron/pending_jobs - pending_jobs cron of the OTRS
# Copyright (C) 2001-2011 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# check every 120 min the pending jobs
45 */2 * * * $HOME/bin/otrs.PendingJobs.pl >> /dev/null
# --
# cron/postmaster - postmaster cron of the OTRS
# Copyright (C) 2001-2011 OTRS AG, http://otrs.org/
# --
# 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.pl >> /dev/null
# --
# cron/postmaster_mailbox - postmaster_mailbox cron of the OTRS
# Copyright (C) 2001-2011 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# fetch emails every 10 minutes
```

```

*/10 * * * * $HOME/bin/otrs.PostMasterMailbox.pl >> /dev/null
# --
# cron/rebuild_ticket_index - rebuild ticket index for OTRS
# Copyright (C) 2001-2011 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# just every day
01 01 * * * $HOME/bin/otrs.RebuildTicketIndex.pl >> /dev/null

# --
# cron/session - delete old session ids of the OTRS
# Copyright (C) 2001-2011 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# delete every 120 minutes old/idle session ids
55 */2 * * * $HOME/bin/otrs.DeleteSessionIDs.pl --expired >> /dev/null

# --
# cron/unlock - unlock old locked ticket of the OTRS
# Copyright (C) 2001-2011 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# unlock every hour old locked tickets
35 * * * * $HOME/bin/otrs.UnlockTickets.pl --timeout >> /dev/null

linux:/opt/otrs/bin#

```

Script 3.19. Crontab file.

3.3. Upgrading the OTRS Framework

These instructions are for people upgrading OTRS from 2.4 to 3.0, and applies both for RPM and source code (tarball) upgrades.

If you are running a lower version of OTRS you have to follow the upgrade path to 2.4 first (1.1->1.2->1.3->2.0->2.1->2.2->2.3->2.4->3.0 ...).

If you need to do a "patch level upgrade", which is an upgrade for instance from OTRS version 3.0.3 to 3.0.4, you should skip steps 8, 9, 10, and 13 - 16.

1. Stop all relevant services

e. g. (depends on used services):

```
shell> /etc/init.d/cron stop
shell> /etc/init.d/postfix stop
shell> /etc/init.d/apache stop
```

2. Backup everything below \$OTRS_HOME (default: OTRS_HOME=/opt/otrs):

- Kernel/Config.pm
- Kernel/Config/GenericAgent.pm
- Kernel/Config/Files/ZZZAuto.pm
- var/*
- as well as the database

3. Make sure that you have backed up everything. ;-)

4. Setup new system (optional). If possible try this install on a separate machine for testing first.

5. Install the new release (tar or RPM)

- With the tarball:

```
shell> cd /opt
shell> tar -xzf otrs-x.x.x.tar.gz
shell> ln -s otrs-x.x.x otrs
```

Restore old configuration files.

- Kernel/Config.pm
- Kernel/Config/GenericAgent.pm
- Kernel/Config/Files/ZZZAuto.pm

- With the RPM:

```
shell> rpm -Uvh otrs-x.x.x.-01.rpm
```

Restore old configuration files is not needed, done by RPM.

6. Own themes

Note: The OTRS themes between 2.4 and 3.0 are *_not_* compatible, so don't use your old themes!
Themes are located under \$OTRS_HOME/Kernel/Output/HTML/*/*.dtt (default:
OTRS_HOME=/opt/otrs)

7. Set file permissions

If the tarball is used, execute:

```
shell> cd /opt/otrs/  
shell> bin/otrs.SetPermissions.pl
```

with the permissions needed for your system setup.

8. Apply the database changes (part 1/2)

```
shell> cd /opt/otrs/  
  
# MySQL:  
shell> cat scripts/DBUpdate-to-3.0.mysql.sql | mysql -p -f -u root otrs  
# PostgreSQL:  
shell> cat scripts/DBUpdate-to-3.0.postgresql.sql | psql otrs
```

9. Run the migration script (as OTRS user, *_not_* as root):

You must execute the migration script to migrate some data from the old database structure to the new one. Please run

```
shell> scripts/DBUpdate-to-3.0.pl
```

10. Apply the database changes (part 2/2):

```
# MySQL:  
shell> cat scripts/DBUpdate-to-3.0-post.mysql.sql | mysql -p -f -u root otrs  
# PostgreSQL:  
shell> cat scripts/DBUpdate-to-3.0-post.postgresql.sql | psql otrs
```

11. Refresh the configuration and delete caches. Please run:

```
shell> bin/otrs.RebuildConfig.pl  
shell> bin/otrs.DeleteCache.pl
```

12. Restart your services

e. g. (depends on used services):

```
shell> /etc/init.d/cron start
shell> /etc/init.d/postfix start
shell> /etc/init.d/apache start
```

13. Check for encoding issues

With OTRS 3.0, the default charset of OTRS was changed from "iso-8859-1" to "utf-8".

This will only affect you if you did not specify a charset in `Kernel/Config.pm` (all installations that were made with the web installer have a custom setting there) or if you didn't change the default charset in the `AdminSysConfig`.

If you experience problems with the new charset, add this line to `Kernel/Config.pm`:

```
$Self->{'DefaultCharset'} = 'iso-8859-1';
```

In general, using "utf-8" is the recommended mode of running OTRS, and switching from "iso-8859-1" should work well.

Please note: We recommend to change existing non-UTF-8 installations of OTRS to UTF-8 with the upgrade to 3.0. All other encodings are now deprecated.

OTRS 3.1 will only allow UTF-8 as internal charset.

14. Starting with OTRS 3.0, application modules may have their own special CSS and/or JavaScript.

This is part of the so-called frontend module registration of the modules in the `SysConfig`. This may cause problems during the upgrade, because locally modified frontend module configuration settings will not catch the updates of the new default configuration.

To check if you are affected, please look at the file `Kernel/Config/Files/ZZZAuto.pm` and look for entries like

```
$Self->{'Frontend::Module'}->{'AgentStats'} = { ... };
```

These settings should be reset to their default value in Admin -> SysConfig (in this case: 'Frontend::Module###AgentStats') with the little "Reset" button next to the "Active" checkbox. Then the settings will be reset with the new module-specific settings and no longer show up in

`Kernel/Config/Files/ZZZAuto.pm`. After this procedure, you may customize these settings again, if needed.

The same procedure must be executed for the setting "Frontend::ToolBarModule###1-Ticket::TicketSearchFulltext", as this now also uses module specific CSS. If this setting is not activated on your system, you can safely omit this step.

15. Customer Database Backend Field Renaming

The field 'salutation' in the built-in customer database has been renamed to the more appropriate 'title'. If you are using the built-in database data source for customers, and you have changed the configuration, for instance because you've added fields to the customer table, or because you've enabled Customer Company support, you should change the mapping in your `Kernel/Config.pm`:

Change this line:

```
[ 'UserSalutation', 'Salutation', 'salutation', 1, 0, 'var', "", 0 ],
```

to:

```
[ 'UserTitle', 'Title', 'title', 1, 0, 'var', "", 0 ],
```

Please note that this can also affect any salutations or other templates where you've used the string `<OTRS*_UserSalutation>`. If you have used this string AND you use the built-in customer table, please replace it with `<OTRS_CURRENT_UserTitle>` instead.

16. Optional: Mark Tickets as Read

In OTRS 3.0 there is a new feature: new tickets and new articles which an Agent did not read yet are highlighted as 'unread'. For all tickets which were created before the upgrade to OTRS 3.0, the 'read' information is missing, and therefore these tickets will be marked as 'unread'. If this bothers you, you can use a script to set all tickets and articles as read for all agents which have read permissions for these tickets. Note that this script may run for a while!

```
shell> bin/otrs.MarkTicketAsSeen.pl
```

17. Well done!

3.4. Upgrading Windows Installer

There's currently no in-place upgrade available for OTRS installations that were done with the Windows Installer. The upgrade process basically consists of backing up the database and the filesystem,

uninstalling OTRS, installing the new version, restoring the database and running the upgrade procedure if needed.

Upgrading is described in FAQ# 4200351

(<http://faq.otrs.org/otrs/public.pl?Action=PublicFAQ;ItemID=351>), and there is also an informative YouTube video (<http://www.youtube.com/watch?v=sf0R-reMTWc>) available.

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, etc.

Soposing 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 (see Figure 4.1 below).

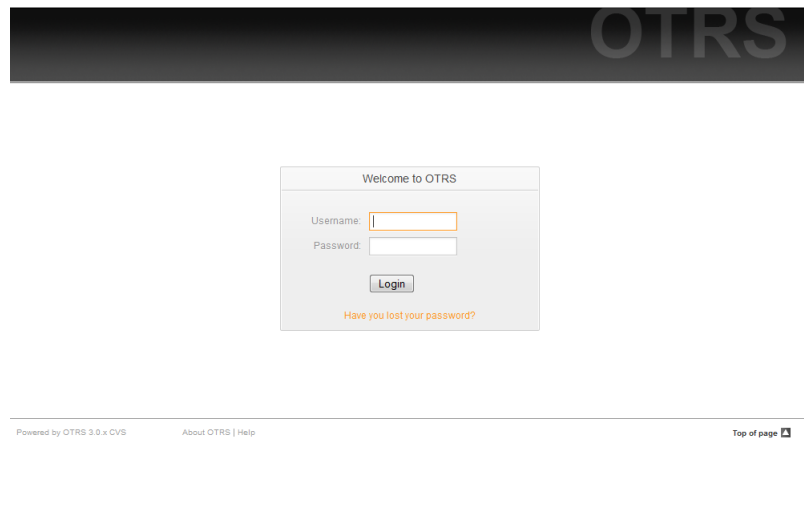


Figure 4.1. Login screen of the agent interface.

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 their account settings, etc.

Supposing that 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 (see Figure 4.2 below).

Company Support

Login

Username Password Login

[Forgot password?](#)

Not yet registered? [Sign up now.](#)

Powered by OTRS 3.0.x CVS

Figure 4.2. Login screen of the customer interface.

4.3. Public web interface

In addition to the web interfaces for agents and customers, OTRS has a public web interface (see Figure 4.3 below) that is available through the FAQ-Module (must be installed separately). It provides public access to the FAQ system, and lets visitors search through the FAQ entries without any authorization.

Supposing that the host with the OTRS installation is reachable via the domain *http://www.example.com* (*http://www.example.com/*) then 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.

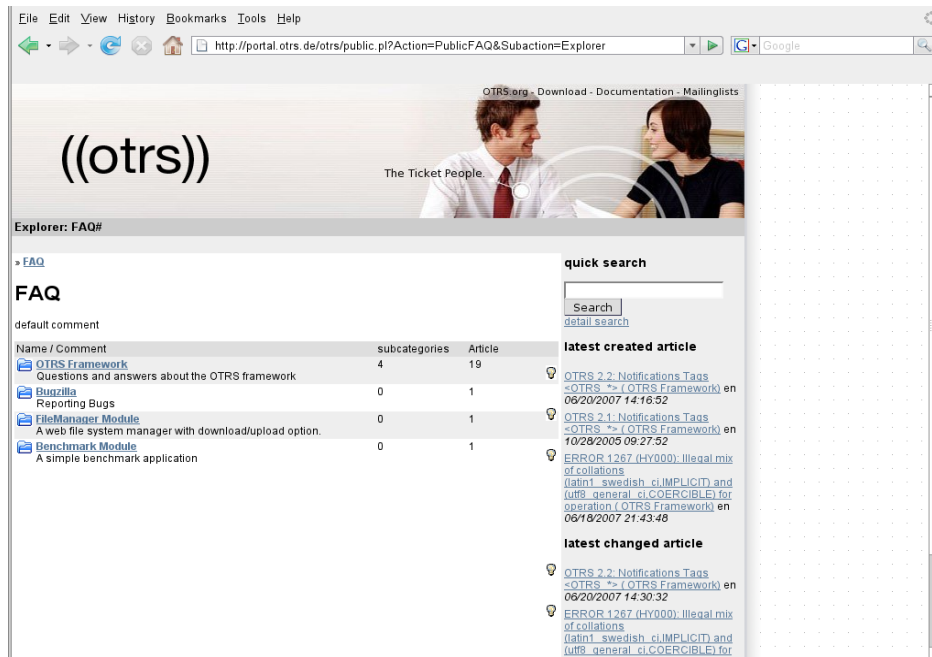


Figure 4.3. Public web interface.

4.4. First login

Access to the login screen, as described in the section about the agent web interface

The login 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, using "root@localhost" for username and "root" for password.

Warning

This account data is valid on every new installed OTRS system. For that reason you should change the password for the OTRS administrator as soon 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.

If you have forgotten your password, you can get a new password through the system. Just press in the link below the login button and enter the mail address that is registered for you OTRS account into the input field and press the submit button (see Figure 4.4).

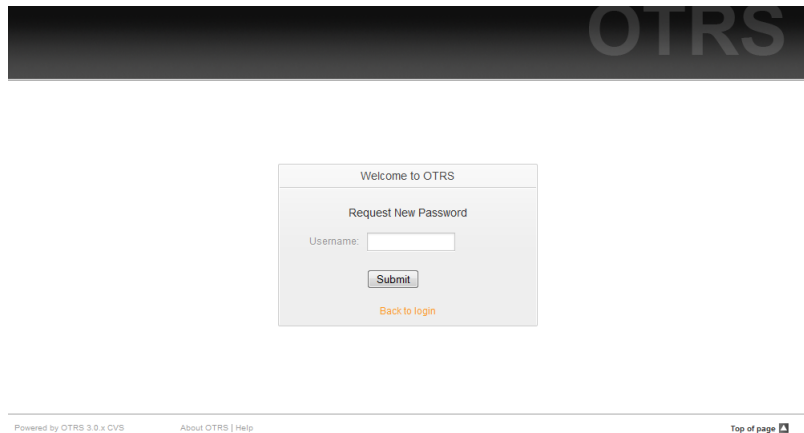
The image shows a screenshot of the OTRS web interface. At the top, there is a dark banner with the word 'OTRS' in large, light-colored letters. Below the banner, the main content area is white. It features a 'Welcome to OTRS' message, followed by a 'Request New Password' section. This section contains a 'Username:' label, a text input field, a 'Submit' button, and a 'Back to login' link. At the bottom of the page, there is a footer with three links: 'Powered by OTRS 3.0.x CVS', 'About OTRS | Help', and 'Top of page' with a small icon.

Figure 4.4. Request new password.

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 Dashboard after the login (see Figure 4.5 below). The Dashboard is completely customizable. At the beginning it shows your locked tickets, allows direct access to the views: queue, status and escalation, and also to create new phone and e-mail tickets. Besides, it gives you a quick summary of the tickets which are: pending, escalated, new and open, among other things.

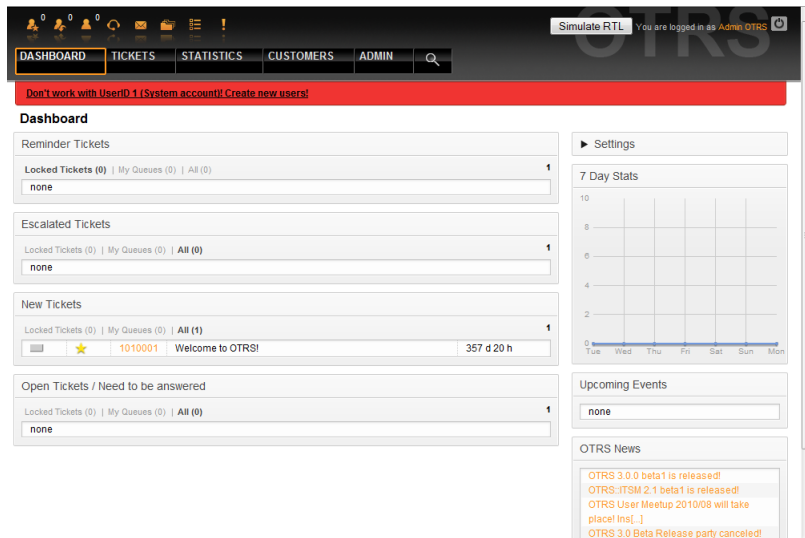


Figure 4.5. Dashboard of the agent interface.

To improve the clarity, the web interface is separated into different areas. At the very top of the windows, there is some general information, like the current username, the logout button, icons that show the amount of locked tickets and allow direct access to them, links to create a new phone/e-mail ticket. It also gives you the change to go to the queue, status and escalation views.

Below the icons is the navigation bar. It shows buttons that enable you to navigate to different areas or modules of the system and let you execute some global actions. By pressing the Dashboard button, you can go to the dashboard (default start page, after login). If you click on the Ticket button, you will get a submenu with the different valid operations, like changing the ticket's view, create a new ticket (phone/e-mail) or search for a specific ticket. The Statistics button is also a menu that allows choosing from having an overview of the registered statistics, create a new one or import an existing one. The Customers button leads you to the Customer Management. By clicking the Admin button, you can access to all the administrator modules, where you can, e.g. create new agents, queues, etc. There is also a Search button to make ticket searches.

If some additional application are installed, e.g. the file manager or the web mailer, buttons to reach this applications are also displayed.

The red 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.

Below the title of the section you are currently in, there are several subsections (each of them in a separate box), which can be dragged and dropped to reallocate them, as long as they remain in the same

column. In order to drag the single elements, you need to place the cursor in the header of the element.

In the left column you can see the data about some tickets (if available), clasified in: reminder, escalated, new and open. In each of the categories, you are also able to see all tickets you are allowed to access or how many tickets you have locked and how many are located in "My Queues".

"My Queues" are the queues, that you want to observe more attentive. You can select them in your account preferences .

In the right column you can find the current settings, which will be expanded/hidden by pressing in the Settings area, as shown in the Figure 4.6 (this section is fixed, so you can not drag and drop it). Once you accessed to the settings, you can check/unchecked the items and save your changes.

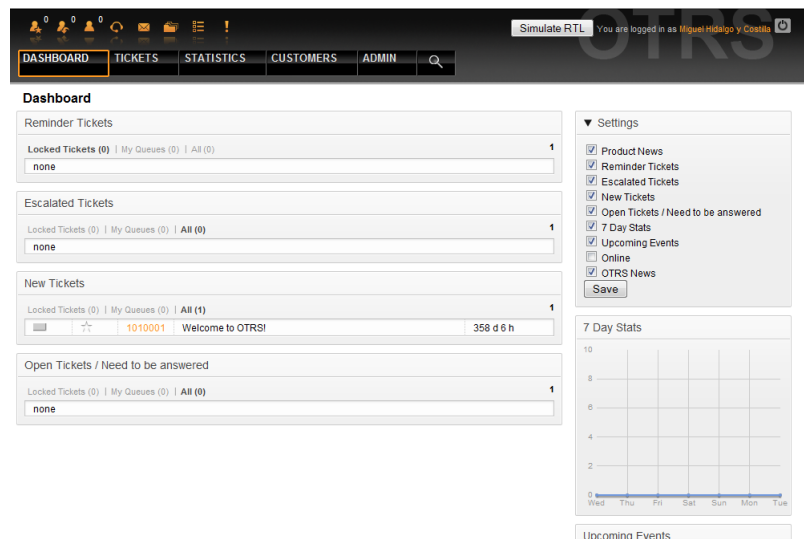


Figure 4.6. Settings of the dashboard.

Below the settings you can see a statistic of the last 7 days' tickets, as you might expect, this graphic will be empty in a brand new system. There is also a section for upcoming events and OTRS news.

On the bottom of the screen the site footer is displayed (see Figure 4.7 below). It contains some links to directly access to the OTRS official website, get information about OTRS, ask for help or go to the top of the page.

Figure 4.7. Footer.

4.6. What is a queue?

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

On common mail systems all messages are into an inbox file. An inbox is a big file where the messages are stored. 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 extra 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 installed OTRS to get rid of his support chaos. His agents and himself are using the system to manage the support requests for the video recorders.

One queue that contains all request is enough for this situation. However, after some time, Max decides to also sell DVD recorders. Now, the customers not only have questions about the video recorder, but also about the new product. More and more emails get into the one queue of Max's OTRS and it's hard to keep the overview.

One day, Max decides to optimize the structure of his support system and adds two new queues, thus now three queues are used. Into the old queue ("raw") all mails are stored if they arrive at the ticket system. The two new queues are one for the video recorder ("video recorder") and the other one for the dvd recorder requests ("dvd recorder").

Max asks Sandra to watch the "raw" queue and sort (dispatch) the mails either into "video recorder" or "dvd recorder", depending on what the customer is asking. John only has access to the "video reorder" 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 to create the structure of a company. Max could implement, for example, another queue called "sales", which could contain the sub queues "requests", "offers", "orders", "billing", etc. 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, consequently less 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 customized to meet each one needs, via the user preferences. Customers, agents and the OTRS administrator have the possibility to adjust their own account preferences. In order to do so, agents have to click on their name at the right top corner of the web interface (see Figure 4.8 below), and customers must click on the "Preferences" link (see Figure 4.9 below).

Figure 4.8. Agent's personal preferences.

An agent can configure 3 different categories of preferences: user profile, email settings and other settings. The default possibilities are:

User Profile

- Change the current password.
- Adjust the interface language.
- Choose the default spelling dictionary.
- Switch the frontend skin to wear.
- Shift the frontend theme.
- Activate and configure the out of office time.

Email Settings

- Select the events when the system shall send a notification to the email address.

Other Settings

- Select the queues you want to monitor in "My Queues".
- Set the refresh period for the queue view.
- Choose the screen that should be displayed after a ticket is created.

Figure 4.8. Customer's personal preferences.

A customer can select the web interface language, the refresh period for the ticket overview, manage the S/MIME certificates and choose the maximum amount of shown tickets. It is also possible to set a new password.

Chapter 5. The admin area of OTRS

5.1. Basics

This is the central place for OTRS administrators. Here you can alter your configuration, install additional packages, such as FAQ and ITSM; add queues and agents, and much more.

Agents who are members of the *admin* group can access the administrators area, by clicking the *Admin* link, in the navigation bar (see Figure 5.1 below); the rest of the agents won't see this link. On a new installation you can log in with the user *root@localhost* and the password *root*.

Warning

Please change the password for this account as soon as possible via the [user preferences](#) page, because this is a default password on all OTRS systems.

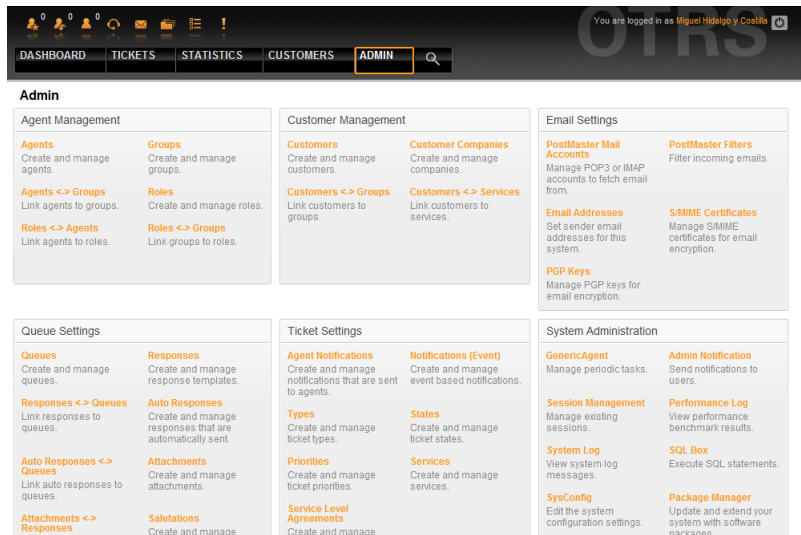


Figure 5.1. Administrator's area of the agent interface.

5.2. Agents, groups and roles

5.2.1. Agents

By clicking the link *Agent* you get access to the agent management of OTRS (see Figure 5.2 below). Administrators can add, change or deactivate agent's accounts. Administrators can also manage agent's preferences, for instance, the language of their interface and notification settings.

Note: To keep the consistency of the data for OTRS it is not possible to delete any agent's account, but only deactivate them. To do so, set them to *invalid* or *invalid-temporarily*.

The screenshot shows the OTRS Admin interface. At the top, there's a navigation bar with links: DASHBOARD, TICKETS, STATISTICS, CUSTOMERS, and ADMIN (highlighted). Below the navigation bar, the 'Agent Management' section is visible. It includes an 'Actions' panel with an 'Add agent' button, a 'Filter for Agents' input field, and a 'Note' section stating: 'Agents will be needed to handle tickets. Attention: Don't forget to add a new agent to groups and/or roles!'. To the right, there's a 'List' table showing the following data:

USERNAME	NAME	EMAIL	LAST LOGIN	VALID	CHANGED	CREATED
fcxavier.mina	Francisco Xavier Mina	fcxavier.mina@freedom.com		valid	08/03/2010 10:38	08/03/2010 10:38
miguel.hidalgo	Miguel Hidalgo y Costilla	miguel.hidalgo@freedom.com	08/03/2010 06:04	valid	08/03/2010 06:01	08/03/2010 06:01
root@localhost	Admin OTRS	root@localhost	08/03/2010 05:59	valid	08/03/2010 05:57	08/03/2010 05:57

At the bottom of the page, it says 'Powered by OTRS 3.0.x CVS' and 'Top of page'.

Figure 5.2. Agent's accounts management.

To register an agent, click on the "Add agent" button, type all the needed data and press the Submit button at the bottom of the screen that is shown in the Figure 5.3.

The screenshot shows the OTRS Admin interface. The top navigation bar has tabs for DASHBOARD, TICKETS, STATISTICS, CUSTOMERS, and ADMIN (which is highlighted). The main content area is titled 'Agent Management' and contains an 'Add Agent' form. The form fields are as follows:

- Title: Mr.
- Firstname: Agustin
- Lastname: de Iturbide
- Username: agustin.deiturbide
- Password: (masked with dots)
- Email: agustin.deiturbide@freedom.com
- Valid: valid (dropdown)
- Language: English (United States) (dropdown)
- Spelling Dictionary: English (dropdown)
- Skin: Default (dropdown)
- Theme: Standard (dropdown)
- Out Of Office Time: On (radio button selected)
- Start: 08 / 03 / 2010

On the left side of the 'Add Agent' form, there is a 'Note' section that says: 'Agents will be needed to handle tickets. Attention: Don't forget to add a new agent to groups and/or roles!'.

Figure 5.3. Adding a new agent.

After the new account has been created, you should make the agent a member of one or more groups or roles. Therefore, you will be redirected automatically to the appropriate screen. For further information about groups and roles, read the Groups and Roles sections of this chapter.

5.2.2. Groups

Groups are very important in OTRS and every agent's account should at least belong to one group or role. In a brand new installation, there are three pre-defined groups available, as it is defined in the Table 5-1.

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

Group	Description
admin	Allowed to perform administrative tasks in the system.
stats	Qualified to access the stats module of OTRS and generate statistics.
users	This is the group where your agents should belong to and have read and write permissions. If agents are in this group and have write rights, they can use all functions of the ticket system.

Note: In a brand new OTRS installation, the group *users* is empty and the agent "root@localhost" belongs per default to the admin and stats groups.

You can access to the group management (see Figure 5.4 below) by clicking the *Groups* link in the admin area.

Group Management

Actions

[Add group](#)

Note

The admin group is to get in the admin area and the stats group to get stats area.
Create new groups to handle access permissions for different groups of agent (e. g. purchasing department, support department, sales department, ...).
It's useful for ASP solutions.

List

NAME	COMMENT	VALID	CHANGED	CREATED
admin	Group of all admins.	valid	08/03/2010 05:57	08/03/2010 05:57
stats	Group for stats access.	valid	08/03/2010 05:57	08/03/2010 05:57
users	Group for default access.	valid	08/03/2010 05:57	08/03/2010 05:57

Powered by OTRS 3.0.x CVS Top of page

Figure 5.4. Group management.

Note: To keep the consistency of the data for OTRS it is not possible to delete groups, but only deactivate them. To do so, set them to *invalid* or *invalid-temporarily*.

To add an agent to a group or to change the agents who belong to a group you can use the link *Agents* <-> *Groups* of the admin area (see Figure 5.5 below).

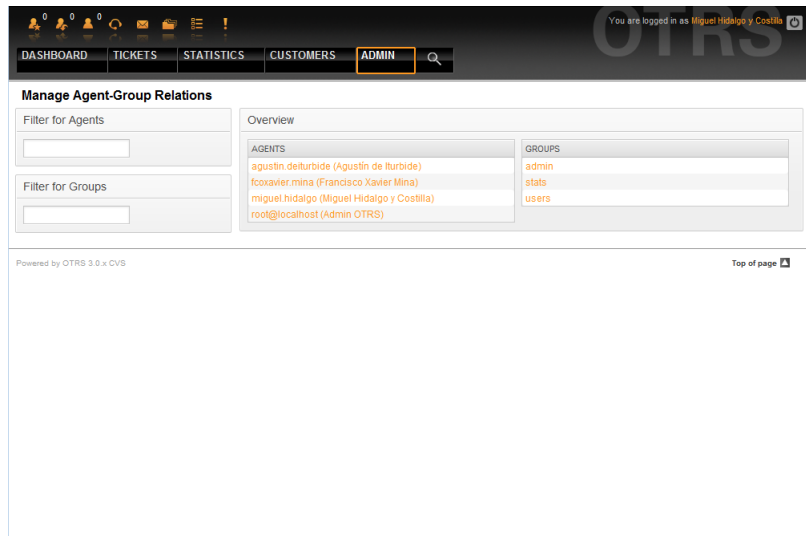


Figure 5.5. Group management.

An overview of all groups and agents in the system is displayed, you can also use the filters to find a specific element. If you want to change the groups that an agent is member of, just click on the agent's name (see Figure 5.6 below). To change the agents that are part of a group just click on the group you want to edit (see Figure 5.7 below).

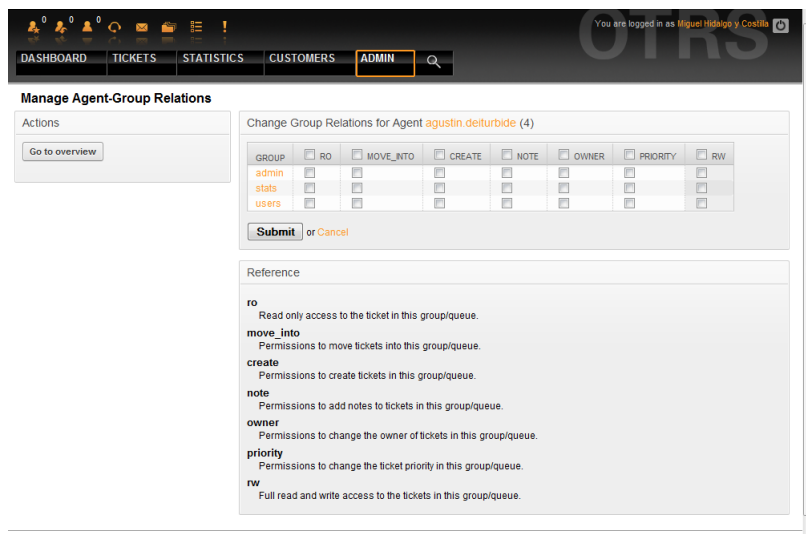
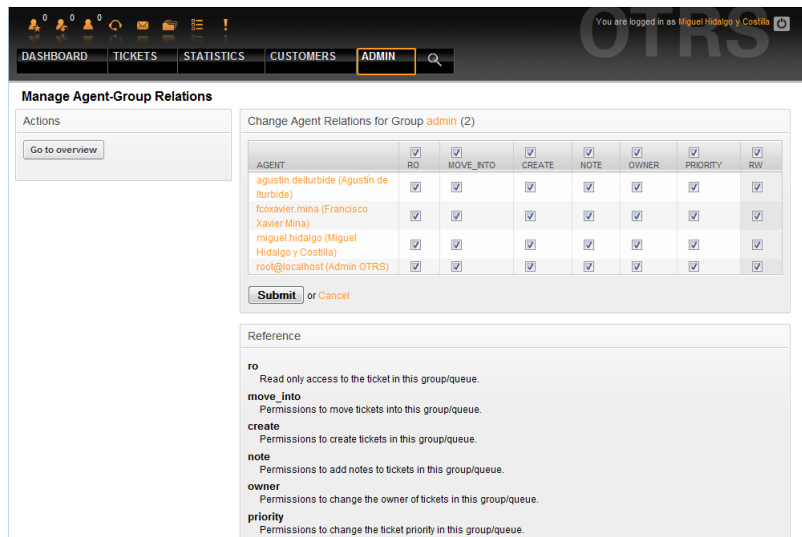


Figure 5.6. Change the groups an agent belongs to.



Manage Agent-Group Relations

Actions

[Go to overview](#)

Change Agent Relations for Group **admin** (2)

AGENT	<input checked="" type="checkbox"/> RO	<input checked="" type="checkbox"/> MOVE INTO	<input checked="" type="checkbox"/> CREATE	<input checked="" type="checkbox"/> NOTE	<input checked="" type="checkbox"/> OWNER	<input checked="" type="checkbox"/> PRIORITY	<input checked="" type="checkbox"/> RW
agustin.deiturbide (Agustín de Iturbide)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
foxavier.mina (Francisco Xavier Mina)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
miguel.hidalgo (Miguel Hidalgo y Costilla)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
root@localhost (Admin OTRS)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Submit](#) or [Cancel](#)

Reference

ro
Read only access to the ticket in this group/queue.

move_into
Permissions to move tickets into this group/queue.

create
Permissions to create tickets in this group/queue.

note
Permissions to add notes to tickets in this group/queue.

owner
Permissions to change the owner of tickets in this group/queue.

priority
Permissions to change the ticket priority in this group/queue.

Figure 5.7. Change the agents that belong to a specific group.

Agents can have different rights in a group, there is a list of the permission rights available in the Table 5-2.

Table 5-2. Rights in the 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 tickets or entries in queues or areas that belong to this group.
rw	Full read and write access on tickets or entries in the queues or areas that belong to this group.

Note: By default, the QueueView only lists tickets in queues that an agent has *rw* access to, i.e., the tickets the agent needs to work on. If you want to change this behaviour, you can set `Ticket::Frontend::AgentTicketQueue###ViewAllPossibleTickets` to `Yes`.

5.2.3. Roles

Roles are a very powerful and helpful feature to manage the access rights of many agents in a very simple and fast way. Therefore, it is ideal to save time on big and complex systems, that have a lot of agents, groups and queues.

To explain the advantages of the role feature, suppose that you have a system with 100 agents, 90 of them have access only to a queue called "support", where all support requests are handled. The "support" queue contains some sub queues. The other 10 agents have permission to access all queues of the system, they are not restricted only to the support queue. These 10 agents 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, etc. must be created and some of the agents in your system shall do this via OTRS. The different agents 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 agents manually, roles that define the different access levels can be created. Then the agents can be added to one or more roles, therefore, their rights will automatically get changed. If a new agent account is created, it is also possible to add this account to one or more roles.

Note: Roles are really useful when maintaining larger OTRS installations. You should not use both Agent to Group and Agent to Role mappings at the same time though, this would make maintenance really hard. Thence, if you decide to go with roles, we'd recommend you to disable the Agents <-> Groups option in the Admin area, by modifying the `Frontend::Module###AdminUserGroup` in the SysConfig. Be aware that this won't remove already existing Agents to Group assignments!

You can access to the role management (see Figure 5.8 below) by clicking the *Roless* link in the admin area.

Role Management

Actions

[Add role](#)

Note
Create a role and put groups in it. Then add the role to the users.

List

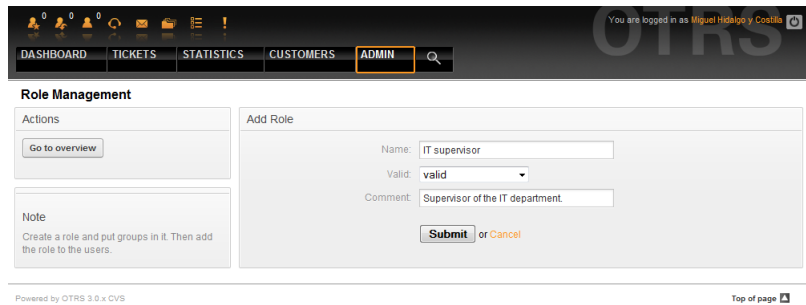
NAME	COMMENT	VALID	CHANGED	CREATED
Help desk	Member of the help desk tel[...]	valid	08/03/2010 14:37	08/03/2010 14:37
IT supervisor	Supervisor of the IT depart[...]	valid	08/03/2010 14:29	08/03/2010 14:29
Service desk	Member of the service desk[...]	valid	08/03/2010 14:37	08/03/2010 14:36

Powered by OTRS 3.0.x CVS Top of page

Figure 5.8. Role management.

Note: To keep the consistency of the data for OTRS it is not possible to delete roles, but only deactivate them. To do so, set them to *invalid* or *invalid-temporarily*.

An overview of all roles in the system is displayed, to edit an entry, just click on the role's name to get its editable data. Please be aware that in a brand new OTRS installation, there are no roles defined by default. To register one, click on the "Add role" button and provide the needed data and submit it (see Figure 5.9 below).



Role Management

Actions

[Go to overview](#)

Note

Create a role and put groups in it. Then add the role to the users.

Add Role

Name:

Valid:

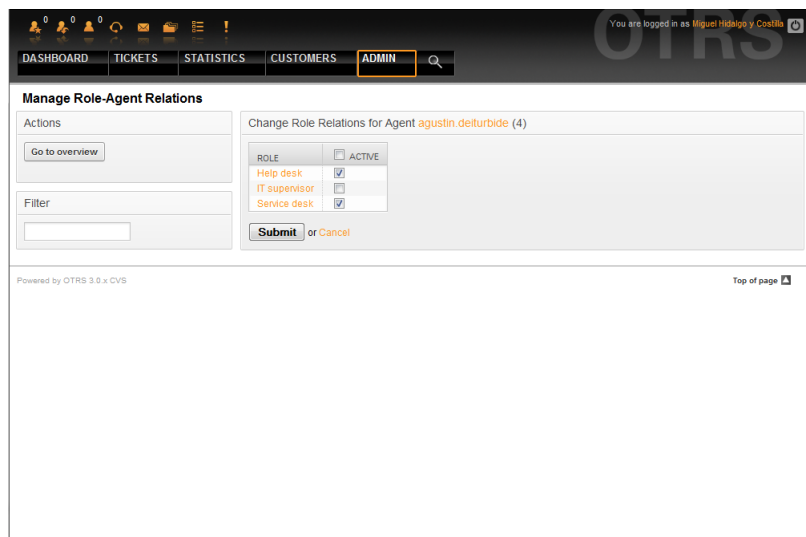
Comment:

or

Powered by OTRS 3.0.x CVS Top of page

Figure 5.9. Adding a new role.

To get an overview of all roles and agents in the system, click on the link Roles <-> Agents in the admin area, you can also use the filters to find a specific element. If you want to change the roles that an agent has, just click on the agent's name (see Figure 5.10 below). To change the agents that have a role just click on the role you want to edit (see Figure 5.11 below).



Manage Role-Agent Relations

Actions

[Go to overview](#)

Filter

Change Role Relations for Agent agustin.deiturbide (4)

ROLE	ACTIVE
Help desk	<input checked="" type="checkbox"/>
IT supervisor	<input checked="" type="checkbox"/>
Service desk	<input checked="" type="checkbox"/>

or

Powered by OTRS 3.0.x CVS Top of page

Figure 5.10. Change the roles an agent has.

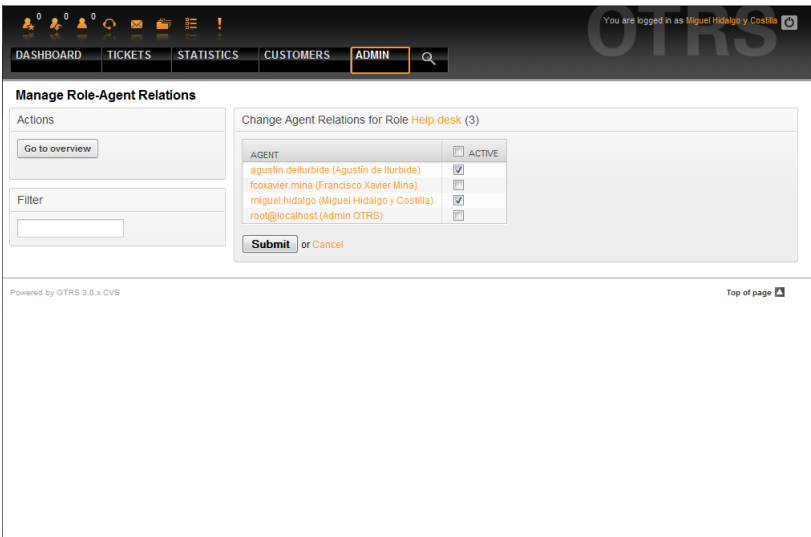


Figure 5.11. Change the agents that have a specific role.

To get an overview of all roles and groups in the system, click on the link Roles <-> Groups in the admin area, you will see a similar screen as the one shown in the Figure 5.12. You can also use the filters to find a specific element.

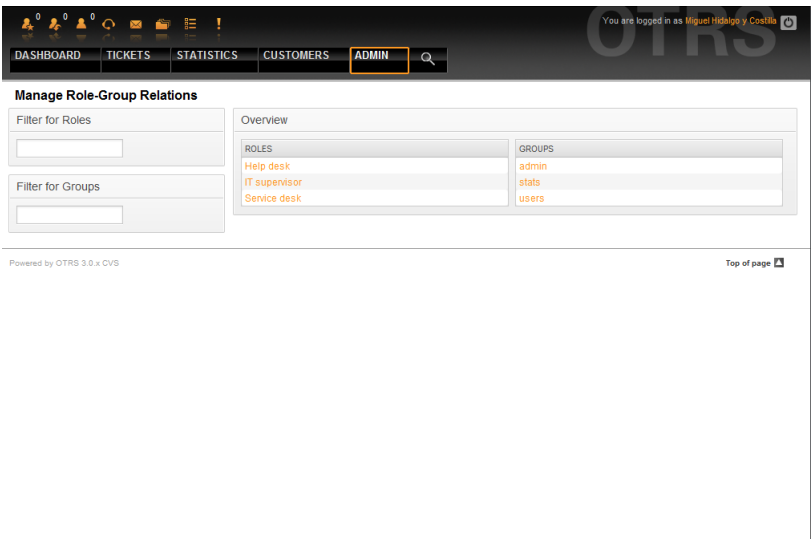


Figure 5.12. Manage role-groups relations.

To define the different access rights for a role, click on the name of a role or a group (see below the Figures 5.13 and 5.14, respectively).

The screenshot shows the OTRS Admin interface. The top navigation bar includes 'DASHBOARD', 'TICKETS', 'STATISTICS', 'CUSTOMERS', and 'ADMIN' (highlighted). The user is logged in as 'Miguel Hidalgo y Costilla'. The main section is titled 'Manage Role-Group Relations'. On the left, there are 'Actions' (a 'Go to overview' button) and a 'Filter' input field. Below these is a 'Note' box explaining that selecting role group permissions is necessary for tickets to be available. The main content area is titled 'Change Group Relations for Role Help desk (3)'. It contains a table with columns: GROUP, RO, MOVE_INT, CREATE, NOTE, OWNER, PRIORITY, and RW. The rows are 'admin', 'stats', and 'users'. The 'admin' row has 'RO' checked. The 'stats' row has 'MOVE_INT' checked. The 'users' row has 'CREATE', 'NOTE', 'OWNER', 'PRIORITY', and 'RW' checked. Below the table are 'Submit' and 'Cancel' buttons. To the right of the table is a 'Reference' section with definitions for 'ro' (Read only access), 'move_into' (Permissions to move tickets), 'create' (Permissions to create tickets), 'priority' (Permissions to change ticket priority), and 'rw' (Full read and write access).

GROUP	RO	MOVE_INT	CREATE	NOTE	OWNER	PRIORITY	RW
admin	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
stats	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
users	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 5.13. Change group relations for a role.

The screenshot shows the OTRS Admin interface. The top navigation bar is the same as in Figure 5.12. The main section is titled 'Manage Role-Group Relations'. On the left, there are 'Actions' (a 'Go to overview' button) and a 'Filter' input field. Below these is a 'Note' box explaining that selecting role group permissions is necessary for tickets to be available. The main content area is titled 'Change Role Relations for Group admin (2)'. It contains a table with columns: ROLE, RO, MOVE_INT, CREATE, NOTE, OWNER, PRIORITY, and RW. The rows are 'Help desk', 'IT supervisor', and 'Service desk'. The 'Help desk' row has 'RO' checked. The 'IT supervisor' row has 'MOVE_INT' checked. The 'Service desk' row has 'CREATE' checked. Below the table are 'Submit' and 'Cancel' buttons. To the right of the table is a 'Reference' section with definitions for 'ro' (Read only access), 'move_into' (Permissions to move tickets), 'create' (Permissions to create tickets), 'priority' (Permissions to change ticket priority), and 'rw' (Full read and write access).

ROLE	RO	MOVE_INT	CREATE	NOTE	OWNER	PRIORITY	RW
Help desk	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
IT supervisor	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Service desk	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 5.14. Change role relations for a group.

5.3. Customers and customer groups

5.3.1. Customers

OTRS supports different types of users. Using the link "Customers" (via the navigation bar or the admin area), you can manage the accounts of your customers (see Figure 5.14 below), who can log into the system via the customer interface (customer.pl). Through this interface, your customers can create tickets and access to them. It is important to know that a customer is needed for the ticket history in the system.

Customer Management

Actions

* Search

Wildcards are allowed.

[Add customer](#)

Database Backend ▾

Note

Customer will be needed to have a customer history and to login via customer panel.

USERNAME	NAME	EMAIL	CUSTOMERID	LAST LOGIN	VALID
ignacio.lopez	Ignacio López Rayón	ignacio.lopez@freedom.com	Independence		valid
leona.vicario	Leona Vicario	leona.vicario@freedom.com	Independence		valid
vicente.guerrero	Vicente Guerrero	vicente.guerrero@freedom.com	Independence	08/03/2010 06:05	valid

Powered by OTRS 3.0.x CVS

Top of page

Figure 5.14. Customer management.

You can search for a registered customer, edit his data by clicking on his name. You also have the possibility to change the customer back-end, for further information please refer to the chapter about external back-ends .

To create a new customer account, click on the "Add customer" button (see Figure 5.15 below). Some of the fields are mandatory, i.e., they have to contain values, so if you leave one of those empty, it will be marked and a message will be displayed.

The screenshot shows the OTRS web interface. At the top, there's a navigation bar with tabs: DASHBOARD, TICKETS, STATISTICS, CUSTOMERS (highlighted), and ADMIN. A user is logged in as 'Miguel Hidalgo y Costilla'. Below the navigation bar, the 'Customer Management' section is active. On the left, there's a sidebar with 'Actions' (Go to overview) and a 'Note' stating: 'Customer will be needed to have a customer history and to login via customer panel.' The main area is titled 'Add Customer' and contains a form with the following fields: Title (Mr.), Firstname (Ignacio), Lastname (Allende), Username (ignacio.allende), Password, Email (ignacio.allende@freedom.com), CustomerID (Independence), Phone, Fax, Mobile, Street, Zip, City, Country, Comment, and a 'Valid' dropdown menu set to 'valid'.

Figure 5.14. Adding a customer.

The customer can access to the system by providing his username and password. The CustomerID is needed by the system to identify the user and his tickets. Since the email address is a unique value, it can be used as ID.

Note: To keep the consistency of the data for OTRS it is not possible to delete customers, but only deactivate them. To do so, set them to *invalid* or *invalid-temporarily*.

5.3.2. Customer groups

Customer users can also be added to a group, which can be useful if you want to add some customers of the same company that shall only have access to one or some queues, through the customer interface. First create the group which your customers will belong to, via the group management module . Then add 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` , via the graphical configuration front-end of OTRS (SysConfig). Using the parameter `CustomerGroupAlwaysGroups` , you can specify the default groups for a new added customer, so that every new account will be automatically added to this groups.

Through the link "Customers <-> Groups" you can manage which customer shall belong to the different groups (see Figure 5.15 below).

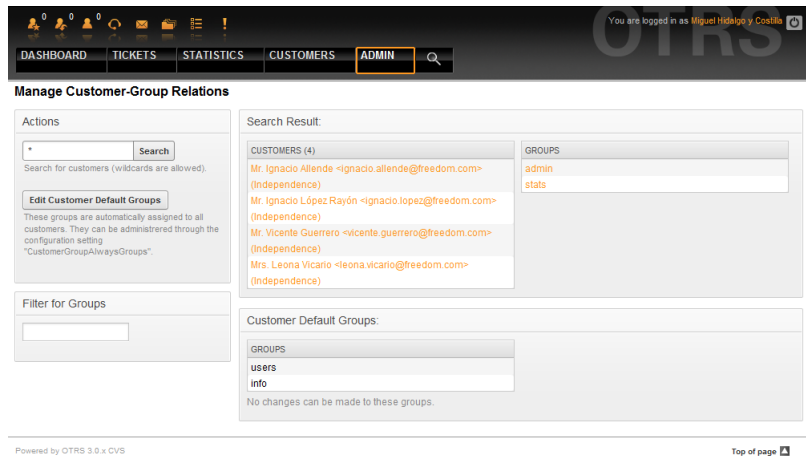


Figure 5.15. Customer-Group relations management.

To define the different groups a customer should be part of and viceversa, click on the corresponding customer username or group (see below the Figures 5.16 and 5.17, respectively).

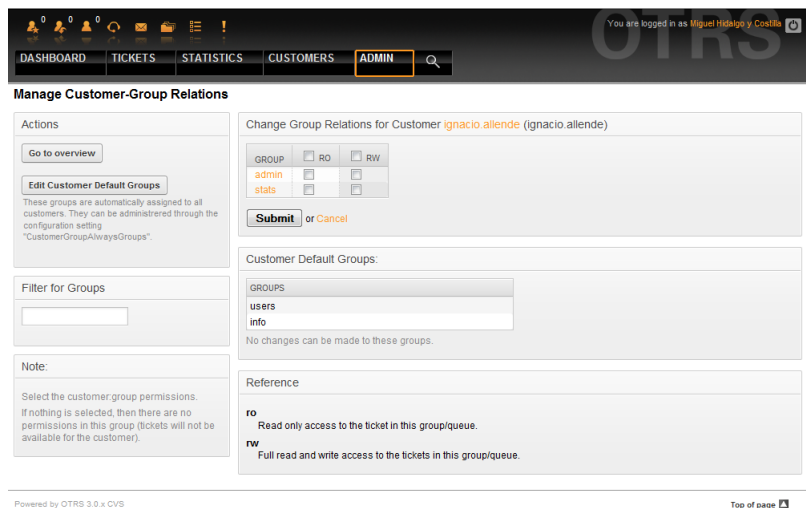


Figure 5.16. Change group relations for a customer.

Manage Customer-Group Relations

Actions

[Go to overview](#)

*

Search for customers (wildcards are allowed).

Note:

Select the customer.group permissions.
If nothing is selected, then there are no permissions in this group (tickets will not be available for the customer).

Change Customer Relations for Group admin (2)

CUSTOMER (4)	RO	RW
Mr. Ignacio Allende <ignacio.allende@freedom.com> (Independence)	<input type="checkbox"/>	<input type="checkbox"/>
Mr. Ignacio López Rayón <ignacio.lopez@freedom.com> (Independence)	<input type="checkbox"/>	<input type="checkbox"/>
Mr. Vicente Guerrero <vicente.guerrero@freedom.com> (Independence)	<input type="checkbox"/>	<input type="checkbox"/>
Mrs. Leona Vicario <leona.vicario@freedom.com> (Independence)	<input type="checkbox"/>	<input type="checkbox"/>

or [Cancel](#)

Reference

ro
Read only access to the ticket in this group/queue.

rw
Full read and write access to the tickets in this group/queue.

Powered by OTRS 3.0.x CVS Top of page

Figure 5.17. Change customer relations for a group.

5.4. Queues

Through the link "Queue" of the admin area, you can manage the queues of your system (see Figure 5.18 below). In a new OTRS installation there are 4 default queues: Junk, Misc, Postmaster and Raw. All incoming messages will be stored in it the "Raw" queue if no filter rules are defined. The "Junk" queue can be used to store spam messages.

Manage Queues

Actions

List

NAME	GROUP	COMMENT	VALID	CHANGED	CREATED
Junk	users	All junk tickets.	valid	08/03/2010 05:57	08/03/2010 05:57
Misc	users	All misc tickets.	valid	08/03/2010 05:57	08/03/2010 05:57
Postmaster	users	Post master queue.	valid	08/03/2010 05:57	08/03/2010 05:57
Raw	users	All default incoming ticket[...]	valid	08/03/2010 05:57	08/03/2010 05:57

Powered by OTRS 3.0.x CVS Top of page

Figure 5.18. Queue management.

Here you can add queues (see Figure 5.19 below) and modify them. You can specify the name and the group that should use the queue; it is also possible to determine if you want it to be a sub-queue of an existing one.

The screenshot shows the OTRS Admin interface. The top navigation bar includes links for DASHBOARD, TICKETS, STATISTICS, CUSTOMERS, and ADMIN. The 'ADMIN' link is highlighted. Below the navigation bar, the 'Manage Queues' section is active. On the left, there is an 'Actions' sidebar with a 'Go to overview' button. The main area is titled 'Add Queue' and contains a form with the following fields and values:

- Name: Support
- Sub-queue of: (empty dropdown)
- Group: admin
- Unlock timeout minutes: 1220
- Escalation - first response time (minutes): 60
- Escalation - update time (minutes): 180
- Escalation - solution time (minutes): 300

Each timeout field has a detailed explanation of its function below it. For example, the 'Unlock timeout minutes' field explains that if an agent locks a ticket and does not close it before the unlock timeout has passed, the ticket will unlock and will become available for other agents.

Figure 5.19. Adding a new queue.

You can define an unlock timeout for a queue, i.e., if an agent locks a ticket and does not close it before the unlock timeout has passed, the ticket will be automatically unlocked and hence, available for other agents.

There are three escalation time settings that can be associated at queue level:

Escalation - First Response Time

- If there is not customer interaction added, either email-external or phone, to a new ticket before the time defined here expires, the ticket is escalated.

Escalation - Update Time

- If there is an article added, such as a follow-up, via e-mail or the customer portal, the escalation update time is reset. If there is no customer contact, either email-external or phone, added to a ticket before the time defined here expires, the ticket is escalated.

Escalation - Solution Time

- If the ticket is not set to closed before the time defined here expires, the ticket is escalated.

With 'Ticket lock after a follow-up' you can define if a ticket should be set to 'locked' to the old owner if a ticket that has been set to closed and later is re-opened. This ensures that a follow up for a ticket is processed by the agent that has previously handled that ticket.

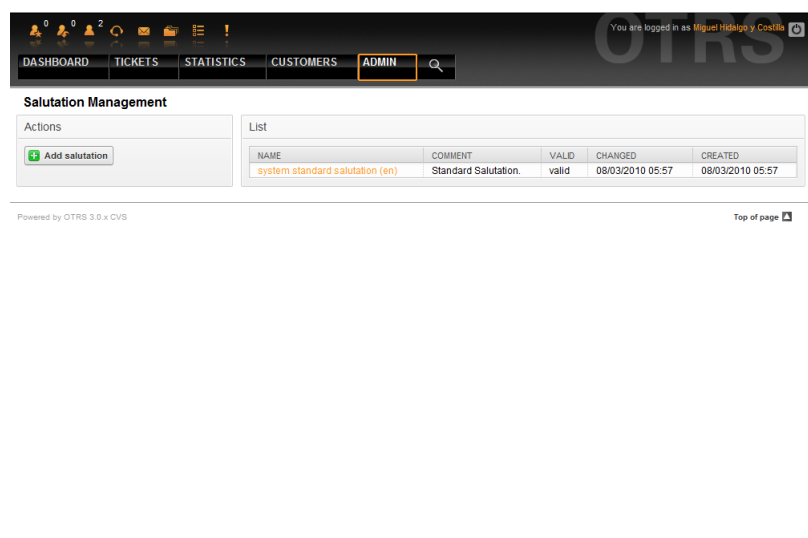
The parameter for the system address specifies the email address that will be used for the outgoing tickets of this queue. There is also possibility to associate a queue to a salutation and a signature, for the email answers. For more detailed information, please refer to the sections [email addresses](#) , [salutations](#) and [signatures](#) .

Note: To keep the consistency of the data for OTRS it is not possible to delete queues, but only deactivate them. To do so, set them to *invalid* or *invalid-temporarily*.

5.5. Salutations, signatures, attachments and responses

5.5.1. Salutations

A salutation is a text module for a response. Salutations can be linked to one or more queues, as described in the section about [queues](#) . A salutation is used only if a ticket from a queue the salutation is linked to, is answered. To manage the different salutations of your system, use the "Salutations" link of the admin area (see Figure 5.20 below). different salutations of your system.



The screenshot shows the OTRS Admin interface. At the top, there's a navigation bar with links: DASHBOARD, TICKETS, STATISTICS, CUSTOMERS, and ADMIN (highlighted). Below the navigation bar, the 'Salutation Management' section is visible. It includes an 'Actions' panel with an 'Add salutation' button. To the right, there's a 'List' table showing the details of existing salutations.

NAME	COMMENT	VALID	CHANGED	CREATED
system standard salutation (en)	Standard Salutation.	valid	08/03/2010 05:57	08/03/2010 05:57

At the bottom of the page, it says 'Powered by OTRS 3.0.x CVS' and 'Top of page'.

Figure 5.20. Salutation management.

After a default installation there is already one salutation available, "system standard salutation (en)".

To create a new salutation, press the button "Add salutation", provide the needed data and submit it (see Figure 5.21 below).

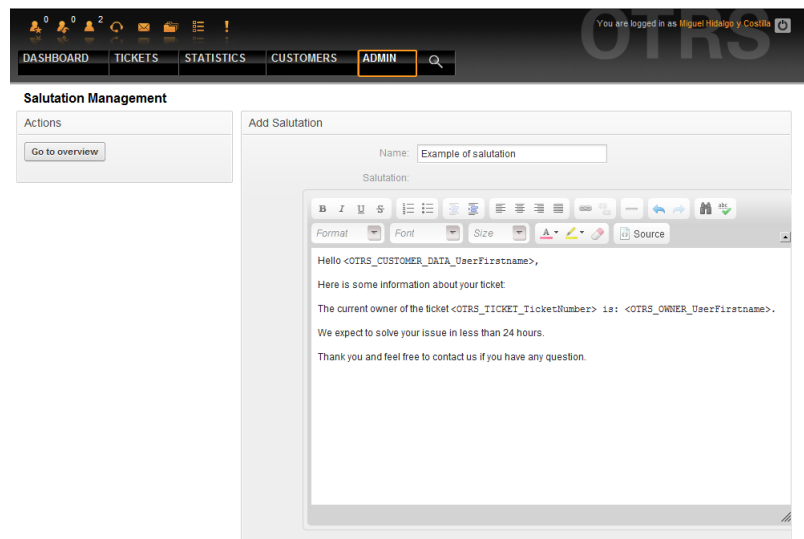


Figure 5.21. Adding a new salutation.

Because the content of a salutation can be created dynamically, for example, for things that change for every ticket (sender's name or email address), it is possible to use variables in salutations. Thus, if you reply a ticket, the variable names will be replaced for their real content, i.e., the text which is saved in the variables.

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 last name of the ticket's sender will be included in your reply.

Note: To keep the consistency of the data for OTRS it is not possible to delete salutations, but only deactivate them. To do so, set them 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 of the admin area, you can manage the signatures in your system (see Figure 5.22 below).

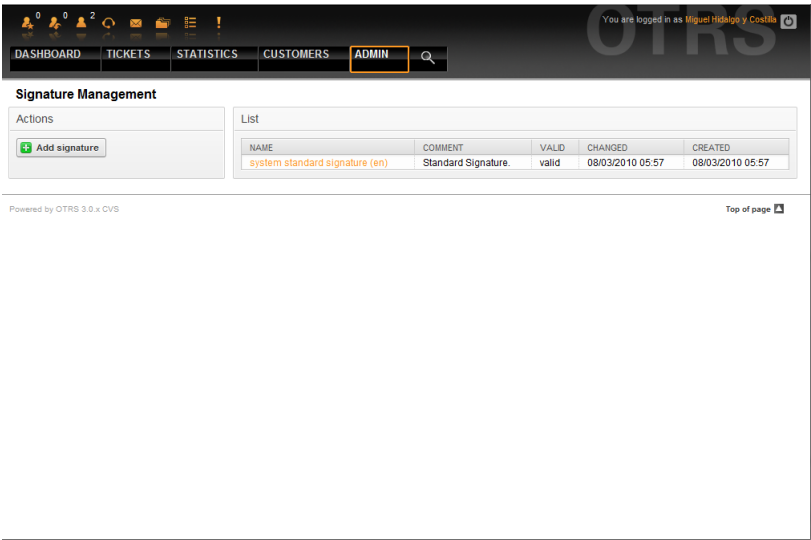


Figure 5.22. Signatures management.

After a default installation of OTRS, there is one predefined signature stored in your system, "system standard signature (en)".

To create a new signature, press the button "Add signature", provide the needed data and submit it (see Figure 5.23 below).

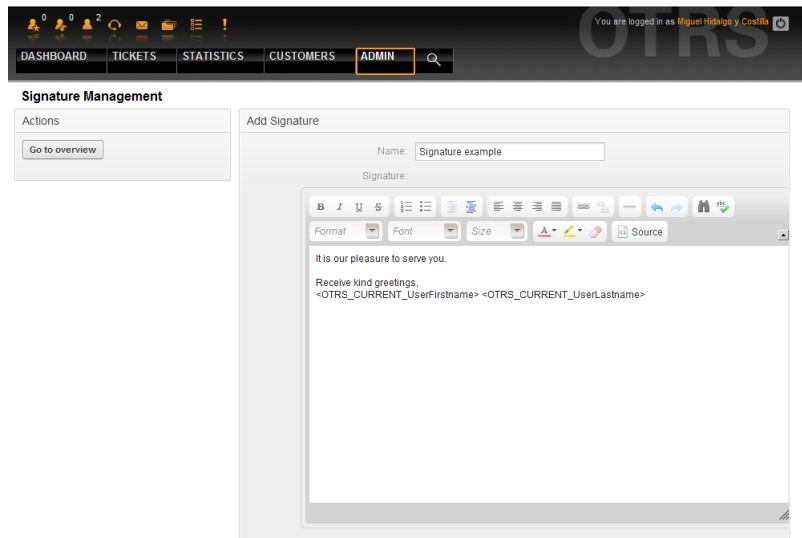


Figure 5.23. Adding a new signature.

Like salutations, signatures can also 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 replace the variable's name.

Note: To keep the consistency of the data for OTRS it is not possible to delete signatures, but only deactivate them. To do so, set them to *invalid* or *invalid-temporarily*.

5.5.3. Attachments

Optionally, you can also create one or more attachments for a response. If the response is selected, the attachments will be attached to the message composition window. If necessary, the agent can remove the attachment from an individual response before sending it to the customer.

Through the "Attachment" link of the admin area, you can load the attachments into the database of the system (see Figure 5.24 below).

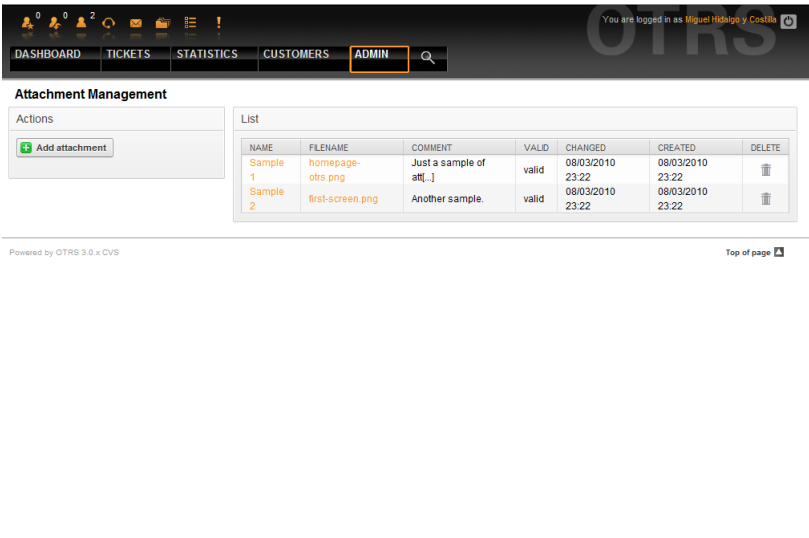


Figure 5.24. Attachments management.

To create a new attachment, press the button "Add attachment", provide the needed data and submit it (see Figure 5.25 below).

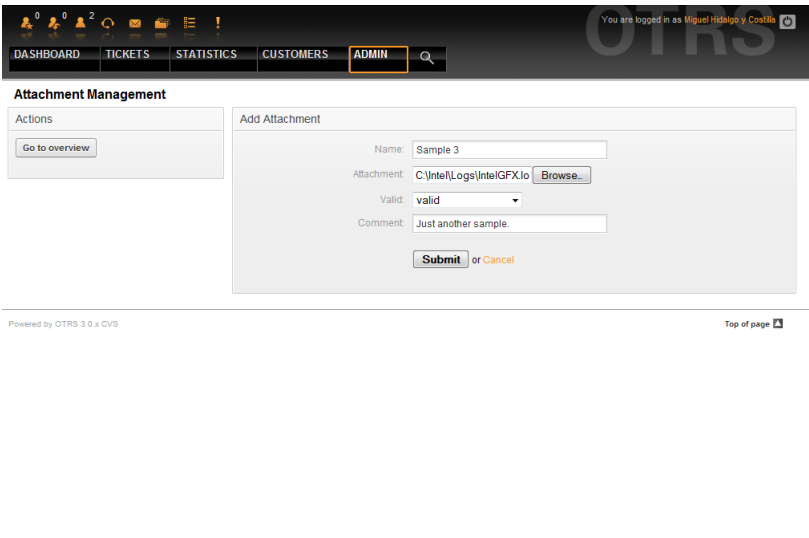


Figure 5.25. Adding a new attachment.

If an attachment is stored it can be linked to one or more responses, just follow the "Attachment <-> Responses" link of the admin area (see Figure 5.26 below).

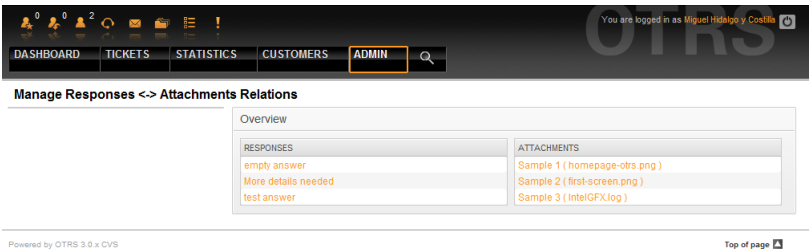


Figure 5.26. Linking attachments to responses.

To define the different attachments a response can have and viceversa, click on the corresponding response name or attachment (see below the Figures 5.27 and 5.28, respectively).

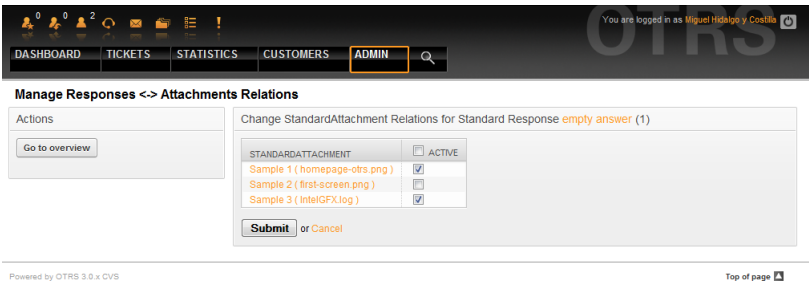


Figure 5.27. Change attachment relations for a response.

0 0 2

DASHBOARD TICKETS STATISTICS CUSTOMERS **ADMIN**

You are logged in as Miguel Hidalgo y Costilla

Manage Responses <-> Attachments Relations

Actions

[Go to overview](#)

Change Standard Response Relations for StandardAttachment **Sample 1** (1)

STANDARD RESPONSE	ACTIVE
empty answer	<input checked="" type="checkbox"/>
More details needed	<input type="checkbox"/>
test answer	<input type="checkbox"/>

[Submit](#) or [Cancel](#)

Powered by OTRS 3.0.x CVS [Top of page](#)

Figure 5.28. Change response relations for an attachment.

5.5.4. 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 and vice versa. In order to be able to use a response quickly, the different responses are displayed below of every ticket in the QueueView or in "My Queues".

After a default installation of the system the "empty answer" response is defined for every queue. Through the "Responses" link in the admin area, it is possible to manage the different responses (see Figure 5.29 below).

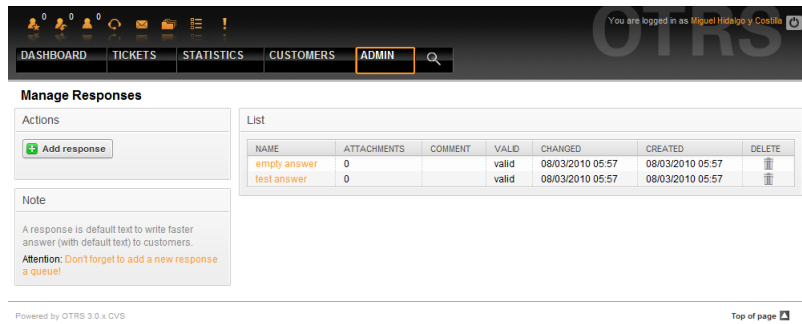


Figure 5.29. Responses management.

To store a new response, just click on the Add response button, provide the required data and submit it (see Figure 5.30 below).

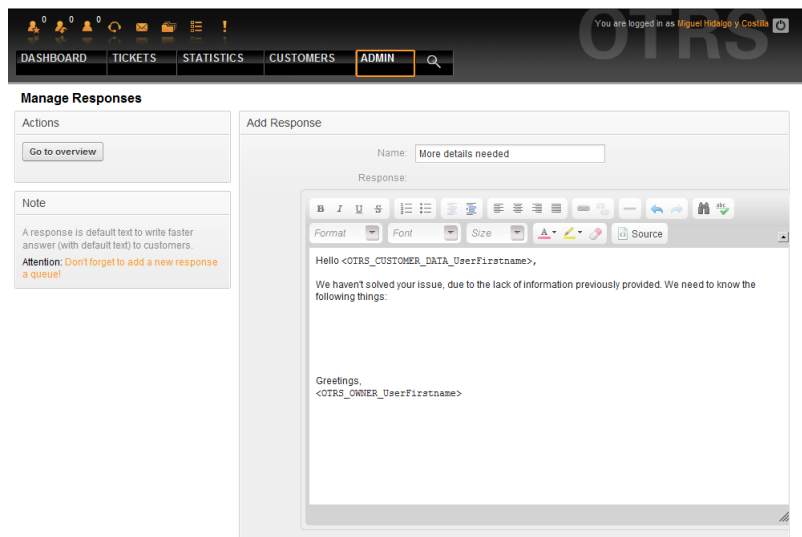


Figure 5.30. Adding a response.

To add/remove responses to one or more queues, the "Responses <-> Queues" link of the admin area can be used (see Figure 5.31 below). You can also use the filters to look for a specific element.

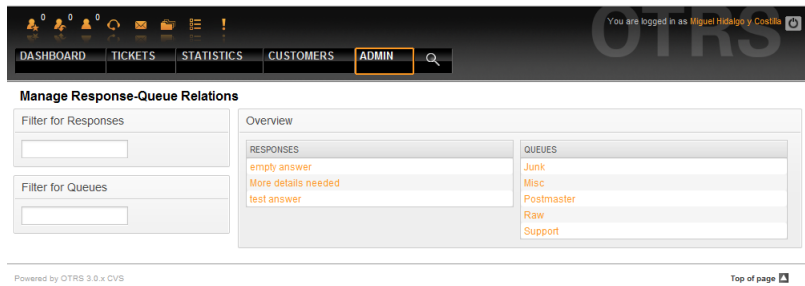


Figure 5.31. Response-Queue relations management.

To define the different responses that will be available for a queue and vice versa, click on the corresponding response or queue (see below the Figures 5.32 and 5.33, respectively).

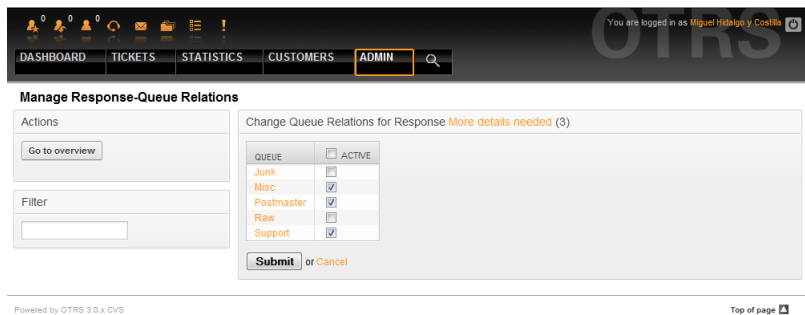


Figure 5.32. Change queue relations for a response.

Manage Response-Queue Relations

Change Response Relations for Queue **Postmaster** (1)

RESPONSE	ACTIVE
empty answer	<input checked="" type="checkbox"/>
More details needed	<input checked="" type="checkbox"/>
test answer	<input type="checkbox"/>

Submit or **Cancel**

Powered by OTRS 3.0.x CVS Top of page

Figure 5.33. Change response relations for a queue.

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 in a way that first the salutation is displayed, then the quoted ticket text, after that, the text of the response comes, and finally, the signature is shown.

5.6. Auto responses

With OTRS, you can send automatic responses to customers if special events happen, e.g. creation of a ticket in certain queue, reception of a follow-up for a ticket, closure or rejection of a ticket, etc. To manage this type of responses, just use "Auto responses" of the admin area (see Figure 5.34 below).

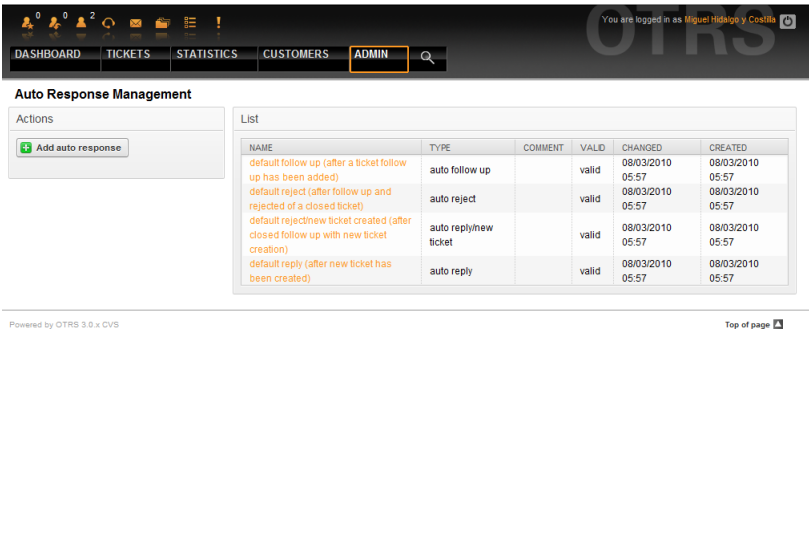


Figure 5.34. Auto response management.

To create an automatic response, click on the button "Add auto response", provide the needed data and submit it (see Figure 5.35 below).

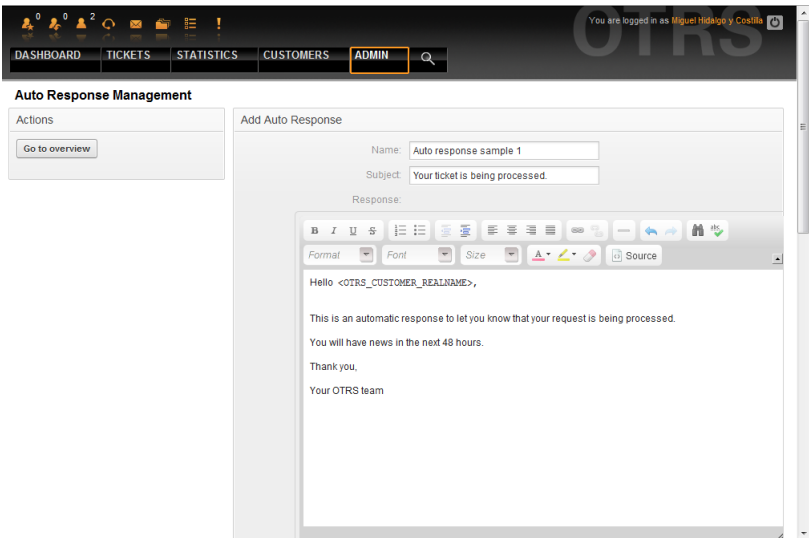


Figure 5.35. Adding an auto response.

The subject and text of auto responses can also be generated by variables, the same as in signatures and salutations. If you insert, for example, the variable <OTRS_CUSTOMER_EMAIL[5]> into the body of the auto answer, the first 5 lines of the customer mail text will be inserted into the auto answer. At the bottom of the screen, shown in the Figure 5.35, you will find more details about the valid variables that can be used.

For every automatic answer, you can specify the event that should trigger it. The system events that are available after a default installation are described in the Table 5-3.

Table 5-3. Events for auto answers

Name	Description
auto reply	Creation of a ticket in a certain queue.
auto reply/new ticket	Reopening of an already closed ticket, e.g. if a customer replies to such ticket.
auto follow up	Reception of a follow-up for a ticket.
auto reject	Automatic rejection of a ticket, done by the system.
auto remove	Deletion of a ticket, done by the system.

Note: To keep the consistency of the data for OTRS it is not possible to delete auto responses, but only deactivate them. To do so, set them to *invalid* or *invalid-temporarily*.

To add an auto response to a queue, use the "Auto Response <-> Queues" link in the admin area (see Figure 5.36 below). All system events are listed for every queue and an auto answer with the same event can be selected or removed via a listbox.

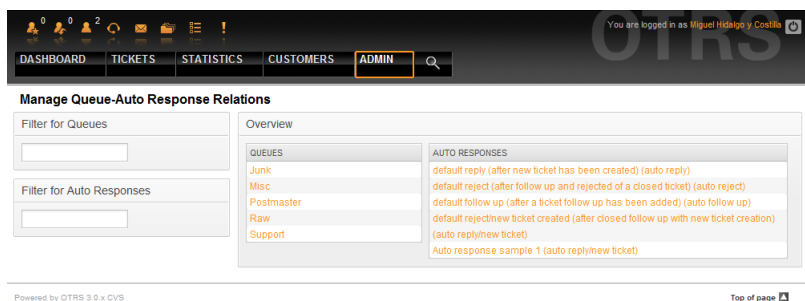


Figure 5.36. Queue-Auto response relations management.

To define the different auto responses that will be available for a queue, click on the corresponding queue name (see Figure 5.37 below). It is also possible to edit an existing auto response, to do so, click on the wished registry and change anything you need to, as if it was a new auto response.

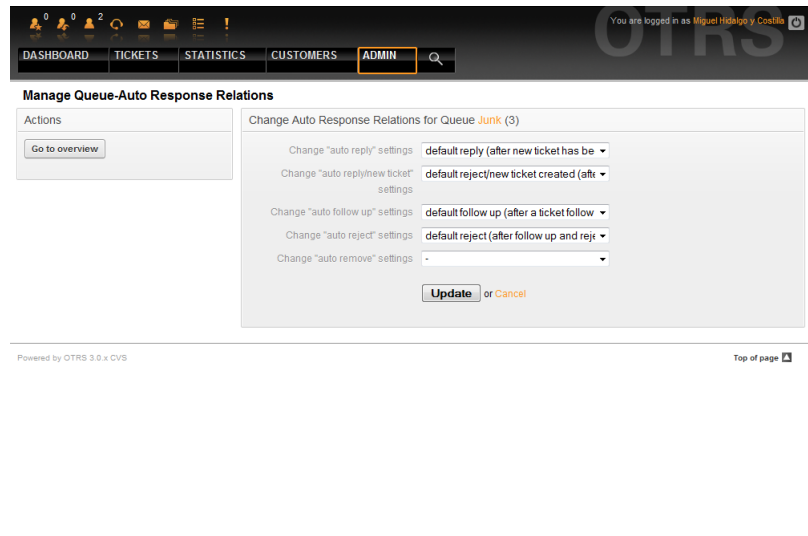


Figure 5.37. Change auto response relations for a queue.

5.7. Email addresses

To enable OTRS to send emails, you need at least a valid email address to be used by the system. Due to a lot of setups need more than one email address, OTRS is able to simultaneously work with several of them. A queue can be linked to many email addresses, and vice versa. The address, that shall be used for outgoing messages from a queue, can be set when the queue is created. Use the "Email Addresses" link of the admin area, to manage all email addresses of the system (see Figure 5.38 below).

System Email Addresses Management

Actions

[Add system address](#)

Note

All incoming email with this address in To or Cc will be dispatched to the selected queue.

List

EMAIL ADDRESS	DISPLAY NAME	COMMENT	VALID	CHANGED	CREATED
otrs@localhost	OTRS System	Standard Address.	valid	08/03/2010 05:57	08/03/2010 05:57
postmaster@mycompany.com	Postmaster team	Example account	valid	08/04/2010 03:52	08/04/2010 03:52
support@mycompany.com	Support team	Example account	valid	08/04/2010 03:51	08/04/2010 03:51

Powered by OTRS 3.0.x CVS Top of page

Figure 5.38. System email addresses management.

If you create a new mail address (see Figure 5.39 below) you can select the queue or sub queue that shall be linked with it. This link enables the system to sort incoming messages via the address in the To: field of the mail into the right queue.

System Email Addresses Management

Actions

[Go to overview](#)

Note

All incoming email with this address in To or Cc will be dispatched to the selected queue.

Add System Email Address

Email address:

Display name:

The display name and email address will be shown on mail you send.

Queue:

Valid:

Comment:

or

Powered by OTRS 3.0.x CVS Top of page

Figure 5.39. Adding a system email address.

Note: To keep the consistency of the data for OTRS it is not possible to delete system email addresses, but only deactivate them. To do so, set them to *invalid* or *invalid-temporarily*.

5.8. Notifications

OTRS allows sending notifications to agents and customer when a specific event happens. Agents can select the system events for their own notifications in the preferences .

Through the "Notification" link in the adminarea, you can manage the notifications of your system (see Figure 5.40 below). With the help of the filters, you can search for specific registries.

The screenshot shows the OTRS Admin interface. The top navigation bar includes 'DASHBOARD', 'TICKETS', 'STATISTICS', 'CUSTOMERS', and 'ADMIN' (selected). The left sidebar shows 'Notification Management' with filters for 'Filter for Language' and 'Filter for Notification'. The main content area displays a table of notifications.

LANGUAGE	NOTIFICATION
Arabic (Saudi Arabia)	Agent-AddNote
Arabic (Saudi Arabia)	Agent-Escalation
Arabic (Saudi Arabia)	Agent-EscalationNotifyBefore
Arabic (Saudi Arabia)	Agent-FollowUp
Arabic (Saudi Arabia)	Agent-LockTimeout
Arabic (Saudi Arabia)	Agent-Move
Arabic (Saudi Arabia)	Agent-NewTicket
Arabic (Saudi Arabia)	Agent-OwnerUpdate
Arabic (Saudi Arabia)	Agent-PendingReminder
Arabic (Saudi Arabia)	Agent-ResponsibleUpdate
Bulgarian (Bългария)	Agent-AddNote
Bulgarian (Bългария)	Agent-Escalation
Bulgarian (Bългария)	Agent-EscalationNotifyBefore
Bulgarian (Bългария)	Agent-FollowUp
Bulgarian (Bългария)	Agent-LockTimeout
Bulgarian (Bългария)	Agent-Move
Bulgarian (Bългария)	Agent-NewTicket
Bulgarian (Bългария)	Agent-OwnerUpdate
Bulgarian (Bългария)	Agent-PendingReminder
Bulgarian (Bългария)	Agent-ResponsibleUpdate
Català	Agent-AddNote
Català	Agent-Escalation
Català	Agent-EscalationNotifyBefore
Català	Agent-FollowUp
Català	Agent-LockTimeout
Català	Agent-Move
Català	Agent-NewTicket

Figure 5.40. Notification management.

You can customize the subject and the text of the notifications. Just select the notification you want to change from the list and load its content by clicking on it (see Figure 5.41). Please note that there is a notification with the same name for each of the available languages.

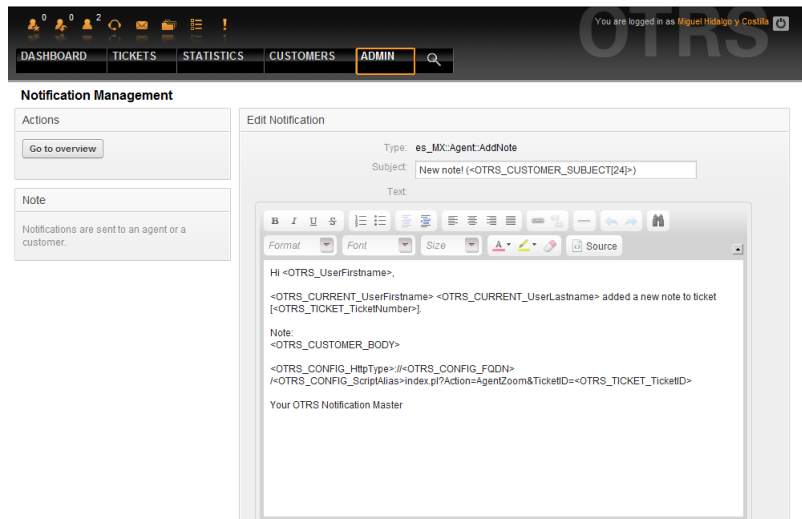


Figure 5.41. Customizing a notification.

Like signatures and salutations, it is possible to dynamically create the content of a notification, by using special variables. At the bottom of the screen shown in the Figure 5.41 you can find a list of them.

It is also possible to create notifications based in events. You can specify in detail when and to whom you want this notification to be sent, since you can choose from a wide diversity of parameters, such as: recipient group(s), agent(s), role(s), email address(es); type of event that will trigger the notification, ticket: type, state, priority, queue, lock, service, SLA, etc.

In order to see a list of all the event based notifications, click on the link Notifications(Event) in the admin area (see Figure 5.42).



Figure 5.42. Event based notification management.

In the screen shown in the Figure 5.42 you can create a new notification by clicking on the Add button (see Figure 5.43).

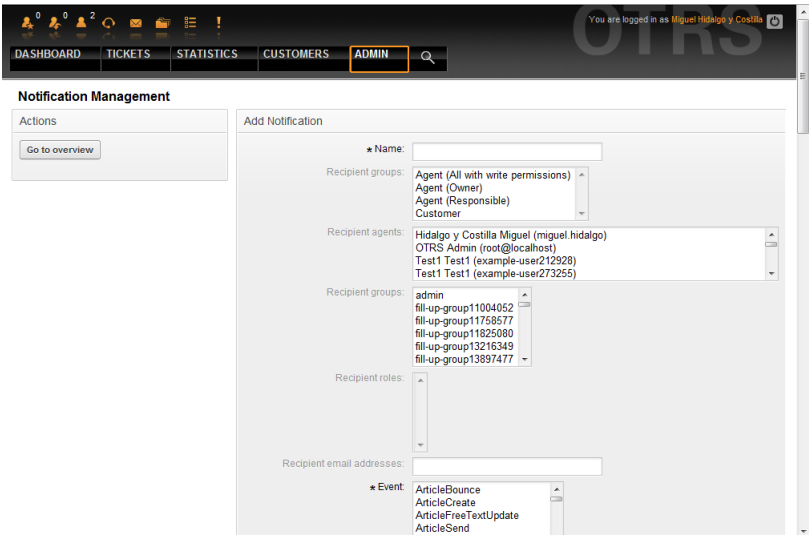


Figure 5.43. Registering an event based notification management.

Please note that the content of the event based notifications can also be dynamically created by using the special variables listed at the bottom of the screen shown in the Figure 5.43.

5.9. SMIME

OTRS can process incoming S/MIME encoded messages and sign outgoing mails. Before this feature can be used, you need to activate it and change some config parameters in the SysConfig.

The "S/MIME Certificates" link in the admin area of OTRS, allows you manage your S/MIME certificates (see Figure 5.44 below). You can add or remove certificates and search through the SMIME data.

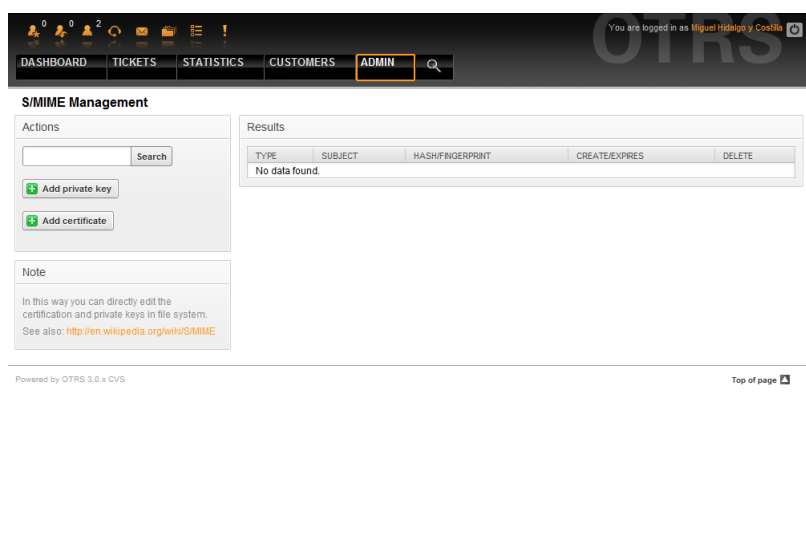


Figure 5.44. S/MIME management.

5.10. PGP

OTRS handles PGP keys, which allows you to encrypt/decrypt messages and to sign outgoing messages. Before this feature can be used, you need to activate it and change some config parameters in the SysConfig.

Through the "PGP Keys" 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 (see Figure 5.45 below), 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.

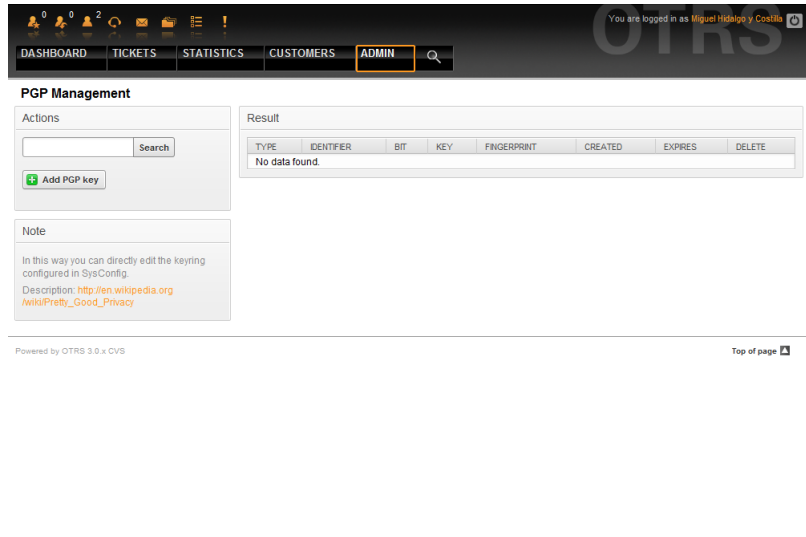


Figure 5.45. PGP management.

5.11. States

Through the "States" link, in the admin area of OTRS, you can manage the different ticket states you want to use in the system (see Figure 5.46 below).

State Management

Actions

[Add state](#)

Note

Attention: Take care that you also updated the default states in you Kernel/Config.pm! See also: <http://doc.otrs.org>

List

NAME	TYPE	COMMENT	VALID	CHANGED	CREATED
closed successful	closed	Ticket is closed suc[...]	valid	08/03/2010 05:57	08/03/2010 05:57
closed unsuccessful	closed	Ticket is closed uns[...]	valid	08/03/2010 05:57	08/03/2010 05:57
merged	merged	State for merged tic[...]	valid	08/03/2010 05:57	08/03/2010 05:57
new	new	New ticket created b[...]	valid	08/03/2010 05:57	08/03/2010 05:57
open	open	Open tickets.	valid	08/03/2010 05:57	08/03/2010 05:57
pending auto close+	pending auto	Ticket is pending fo[...]	valid	08/03/2010 05:57	08/03/2010 05:57
pending auto close-	pending auto	Ticket is pending fo[...]	valid	08/03/2010 05:57	08/03/2010 05:57
pending reminder	pending reminder	Ticket is pending fo[...]	valid	08/03/2010 05:57	08/03/2010 05:57
removed	removed	Customer removed tic[...]	valid	08/03/2010 05:57	08/03/2010 05:57

Powered by OTRS 3.0.x CVS

Top of page

Figure 5.46. State management.

After a default setup, there are some states defined:

- closed successful
- closed unsuccessful
- merged
- new
- open
- pending auto close+
- pending auto close-
- pending reminder
- removed

Every state is linked to a type, which needs to be specified if a new state is created. Per default the state types are:

- closed
- merged
- new
- open
- pending auto
- pending reminder

- removed

5.12. SysConfig

The SysConfig is the place where many configuration options for OTRS are stored.

The "SysConfig" link in the admin area of OTRS loads the graphical configuration front-end (see Figure 5.47 below). 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 perform a full-text search through all the config parameters.

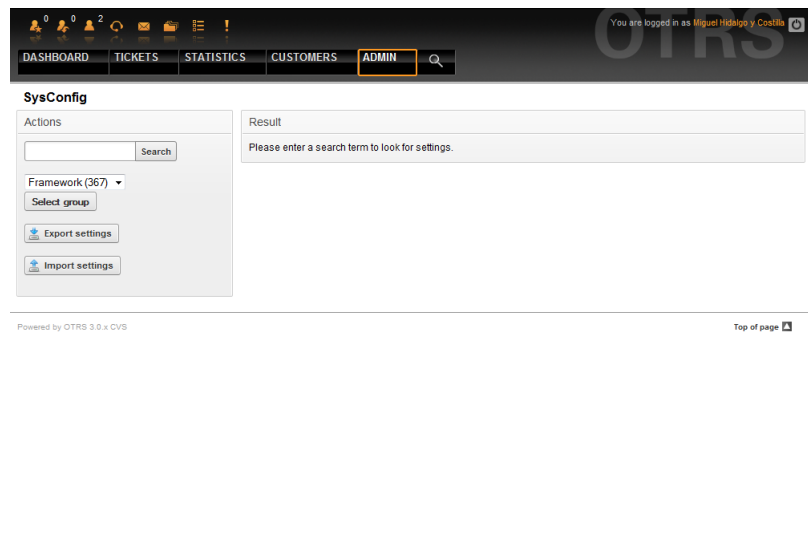


Figure 5.47. The graphical configuration front-end of OTRS (SysConfig).

The graphical configuration front-end is described in more detail in the chapter "Configuring the system through the web interface" .

5.13. Using mail accounts

There are several possibilities to transport new emails into the ticket system. One of them is the

otrs.PostMaster.pl module that pipes the mails directly into the system. Another possibility is the use of mail accounts, which can be administrated through the web interface. The "PostMaster Mail Accounts" link in the admin area of OTRS, loads the management console for the mail accounts (see Figure 5.48 below). OTRS supports the mail protocols: POP3, POP3S, IMAP and IMAPS.

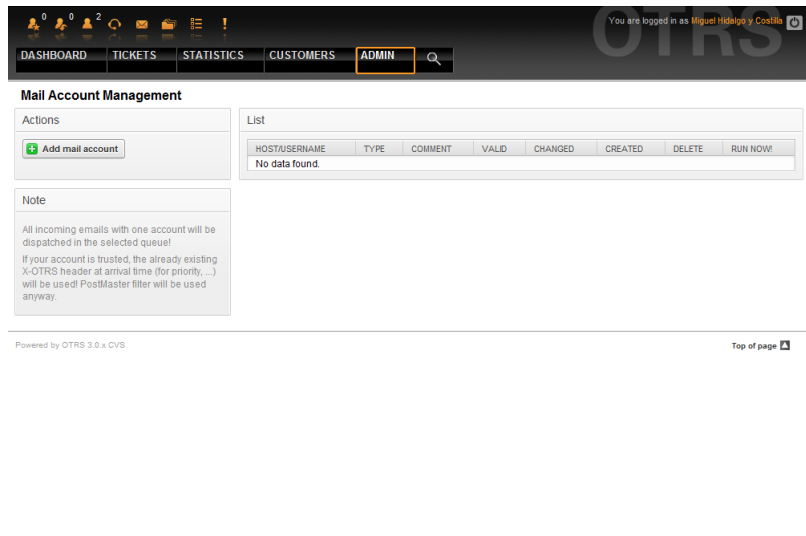


Figure 5.48. Mail account management.

See the section about the PostMaster mail 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 PostMaster.pl or mail accounts are used to get messages into the ticket system. Filter rules can be created through the link "PostMaster Filter" in the admin area of OTRS (see Figure 5.49 below).

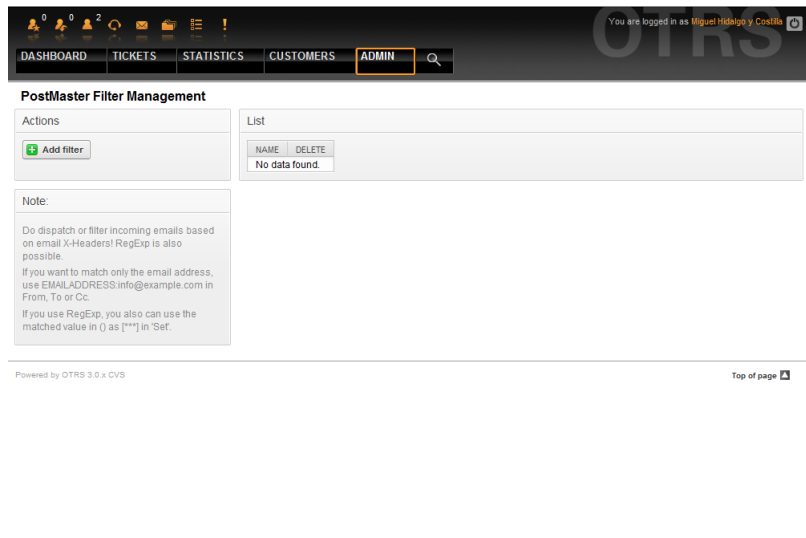


Figure 5.49. PostMaster filter management.

A filter consists of one or more criteria that must match for the defined actions to be executed. You can define filter criteria for the headers or the body of an email, e.g. search for specific header entries or strings in the body, even regular expressions are allowed. All actions for a filter rule are triggered by X-OTRS headers, which are inserted if the filter criteria match. The ticket system evaluates the inserted X-OTRS headers and executes the specific actions. X-OTRS headers can be used to sort an incoming message into a specific queue, change the priority of the message or ignore the message and deliver it not to the system. The Table 5-4 lists the different X-OTRS headers and their meaning.

Note: You also can use X-OTRS-FollowUp-* headers for follow up emails.

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.
X-OTRS-Lock:	lock, unlock	Sets the lock state of a ticket.

Name	Possible values	Description
X-OTRS-Ignore:	Yes or True	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-State-PendingTime:	e. g. 2010-11-20 00:00:00	Sets the pending time of a ticket (you also should sent a pending state via X-OTRS-State).
X-OTRS-Type:	default (depends on your setup)	Sets the type of a ticket (if Ticket::Type support is active).
X-OTRS-Service:	(depends on your setup)	Sets the service of a ticket (if Ticket::Service support is active).
X-OTRS-SLA:	(depends on your setup)	Sets the SLA of a ticket (if Ticket::Service support is active).
X-OTRS-CustomerUser:	CustomerUser	Sets the customer user for the ticket.
X-OTRS-CustomerNo:	CustomerNo	Sets the customer ID for this ticket.
X-OTRS-ArticleKey(1 2 3):	Additional info key for the article.	Saves an additional info key for this article.
X-OTRS-ArticleValue(1 2 3):	Additional info value for the article.	Saves an additional info value for the article.
X-OTRS-SenderType:	agent, system, customer	Sets the type of the ticket sender.
X-OTRS-ArticleType:	email-external, email-internal, email-notification-ext, email-notification-int, phone, fax, sms, webrequest, note-internal, note-external, note-report	Sets the article type for the incoming ticket.
X-OTRS-TicketKey(1 2 ... 8):	Additional info key for the ticket.	Saves an additional info key for the ticket.
X-OTRS-TicketValue(1 2 ... 8):	Additional info value for the ticket.	Saves an additional info value for the ticket.
X-OTRS-Loop:	True	If this X-OTRS header is set no auto answer is delivered to the sender of the message (mail loop protection).

A name must be specified for every filter rule. In the section for "Match" the filter criteria can be

specified. Choose via the listboxes for "Header 1", "Header 2" and so on the parts of the messages where you would like to search and specify via the input fields on the right sides the values. In the section for "Set" you can choose the actions that are triggered, if the filter rules match. You can select for "Header 1", "Header 2" and so on the X-OTRS-Header and set the values (see Figure 5.50 below).

The screenshot shows the OTRS Admin interface. At the top, there's a navigation bar with tabs: DASHBOARD, TICKETS, STATISTICS, CUSTOMERS, and ADMIN (highlighted). A search bar is next to the ADMIN tab. Below the navigation bar, the 'PostMaster Filter Management' section is active. It contains a 'Go to overview' button and a 'Note' section with instructions on how to use filters. The main form is titled 'Add PostMaster Filter'. It has a 'Filtername' field with the value 'My sample filter'. Below it is a 'Stop after match' dropdown set to 'No'. The 'Filter Condition' section has four header fields (Header 1 to Header 4) with dropdown menus and input fields. Header 1 is set to 'From' and the input field contains '.@independence.com'. The 'Set Email Headers' section also has four header fields (Header 1 to Header 4) with dropdown menus and input fields. Header 1 is set to 'X-OTRS-Queue' and the input field contains 'Special queue'.

Figure 5.50. Add a PostMaster filter.

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

A useful filter rule could be to let OTRS automatically sort mails marked for spam with a spam detection tool, such as SpamAssassin, into the "Junk" queue. SpamAssassin adds the "X-Spam-Flag" header to every checked mail. When the mail is marked as spam, the Header is set to "Yes". So the filter criteria would be "X-Spam-Flag: Yes". To create a filter rule with this criteria you can insert the name for example "spam-mails". Also choose in the section for "Match" for "Header 1" from the listbox "X-Spam-Flag:". Insert "Yes" as value for this header. Now the filter criteria is specified. To make sure, that all spam mails get sorted into the "Junk" queue choose in the section for "Set" for "Header 1" the "X-OTRS-Queue:" entry. Specify "Junk" as value for this header. Finally add the new filter rule to activate it for the next new messages in the system.

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

5.15. Executing automated jobs with the GenericAgent

The GenericAgent is a tool to execute tasks automatically, that otherwise should be done by a human person, a real agent. The GenericAgent, for example, can close or move tickets, send notifications on escalated tickets, etc.

To access to the GenericAgent, the link "GenericAgent" in the admin area of OTRS can be used (see Figure 5.51 below). A table with the already created jobs is displayed, where jobs can be executed manually or removed.

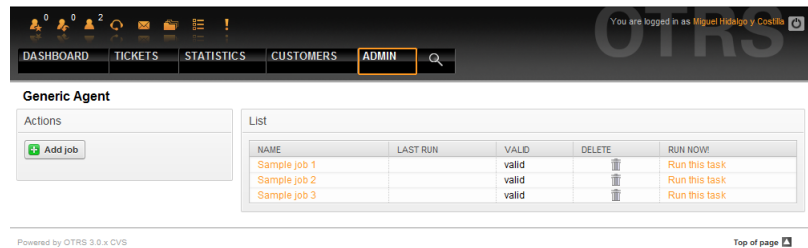


Figure 5.51. Job list for the GenericAgent.

To create a new job a name must be specified and the "Add job" button must be pressed. Via the screen for the creation of a new job for the GenericAgent you can specify the times when the job shall be executed, different criteria to select the tickets to work on and the new properties of those tickets (see Figure 5.52 below).

The screenshot shows the OTRS Admin interface. At the top, there's a navigation bar with tabs: DASHBOARD, TICKETS, STATISTICS, CUSTOMERS, and ADMIN (which is highlighted). A search bar is also present. Below the navigation bar, the 'Generic Agent' section is active. On the left, there's an 'Actions' panel with a 'Go to overview' button. The main area is titled 'Job Settings' and contains the following fields:

- * Job name:** Sample job 4
- Valid:** Yes (dropdown)
- Schedule minutes:** 00, 10, 20 (selected), 30, 40, 50 (dropdown)
- Schedule hours:** 00, 01, 02, 03, 04, 05 (selected) (dropdown)
- Schedule days:** Fri, Mon, Sat (selected), Sun, Thu, Tue, Wed (dropdown)

At the bottom of the form, there is a red warning message: "Currently this generic agent job will not run automatically. To enable automatic execution select at least one value from minutes, hours and days!"

Figure 5.52. Creating a job for the GenericAgent.

If the creation of the job is finished all affected tickets by the job are listed. This list helps you verify that the job is having the expected behavior, no changes are made to these tickets yet. The job will be really activated if it is saved into the job list.

5.16. Admin email

OTRS administrators can send messages to specific users or groups. The "Admin Notification" in the admin area, opens the screen where the agents and groups that should be notified can be selected (see Figure 5.53 below).

Admin Notification

Send Administrative Message to Agents

From:

Send message to users:

Send message to group members:

Group members need to have permission: ☒ ro ☐ rw

Send message to role members:

Also send to customers in groups: ☐

* Subject:

Body:

Figure 5.53. Admin notification.

It is possible to specify the sender, subject and body text of the notification. You can also select the agents, groups and roles who should receive the message.

5.17. Session management

You can see all logged in users and their session details by clicking the "Session Management" link in the admin area (see Figure 5.54 below).

0

0

2

🔄

📁

📁

📁

⚙️

!

DASHBOARD

TICKETS

STATISTICS

CUSTOMERS

ADMIN

🔍

You are logged in as Miguel Hidalgo y Costilla

OTRS

Session Management

Actions

All sessions: 3

Agent sessions: 1

Customer sessions: 2

Unique agents: 1

Unique customers: 2

Kill all sessions

List

SESSION	TYPE	USER	KILL
105b77fe2de2a3d4f78ed780bc8817717	Agent	Miguel Hidalgo y Costilla	Kill this session
10848c1d962f41ef0b5e9d8468b5d037	Customer	Leona Vicario	Kill this session
10e5cdc5fb565e19ac0765f98e9fb5941	Customer	Ignacio López Rayón	Kill this session

Powered by OTRS 3.9.x CVS

Total of page 1

Figure 5.54. Session management.

Some statistics about all active sessions are displayed, e.g. how many agents and customer users are logged in, number of active sessions. Every individual session can be removed by clicking on the *Kill this session* in the right-hand side of the list. You also have the option to *Kill all sessions*, which can be useful if you'd like to bring the system down. Detailed information for every session is available, too (see Figure 5.55 below).

0

0

1

2

You are logged in as Miguel Hidalgo y Costilla

DASHBOARD

TICKETS

STATISTICS

CUSTOMERS

ADMIN

Session Management

Actions

Go to overview

Kill this session

Detail View for SessionID : 105b77fe2de2a3d4f78ed8780bc6817717 - Miguel Hidalgo y Costilla

KEY	VALUE
ChangeTime	2010-08-03 06:01:50
CreateTime	2010-08-03 06:01:50
OutOfOfficeEndDay	4
OutOfOfficeEndMonth	8
OutOfOfficeEndYear	2010
OutOfOfficeStartDay	3
OutOfOfficeStartMonth	8
OutOfOfficeStartYear	2010
SessionID	105b77fe2de2a3d4f78ed8780bc6817717
UserChallengeToken	443f05769205c0299e0c213b1b072eb5
UserEmail	miguel.hidalgo@freedom.com
UserFirstName	Miguel
UserID	2
UsersGroupRo[admin]	Yes
UsersGroupRo[stats]	Yes
UsersGroupRo[users]	Yes
UsersGroup[admin]	Yes
UsersGroup[stats]	Yes
UsersGroup[users]	Yes
UserLanguage	en
UserLastLogin	1280935411
UserLastLoginTimestamp	2010-08-04 10:23:31
UserLastPw	xxxxxxxx
UserLastRequest	1280935483
UserLastName	Hidalgo y Costilla
UserLogin	miguel.hidalgo
UserPw	mmg00r

Figure 5.55. Session details.

5.18. System Log

The "System Log" link in the admin area of OTRS, shows the last log entries of the system (see Figure 5.56 below).

TIME	PRIORITY	FACILITY	MESSAGE
Wed Aug 4 10:23:51 2010	notice	OTRS-CGI-10	Removed SessionID 10b2c771b4115052ca65a75596997c7b3.
Wed Aug 4 10:23:31 2010	notice	OTRS-CGI-10	User: miguel.hidalgo authentication ok (REMOTE_ADDR: 192.168.56.1).
Wed Aug 4 10:23:12 2010	notice	OTRS-CGI-10	Removed SessionID 102d146b9b92158482b556deb742695ab7.
Wed Aug 4 10:12:36 2010	notice	OTRS-CGI-10	CustomerUser: ignacio.lopez Authentication ok (REMOTE_ADDR: 192.168.56.1).
Wed Aug 4 10:12:17 2010	notice	OTRS-CGI-10	CustomerUser: leona.vicario Authentication ok (REMOTE_ADDR: 192.168.56.1).
Wed Aug 4 10:11:05 2010	notice	OTRS-CGI-10	Removed SessionID 107bdfaba2f0ed4abe496b1f6f9fc31921.
Wed Aug 4 10:11:05 2010	notice	OTRS-CGI-10	User: agustin.delturbide authentication ok (REMOTE_ADDR: 192.168.56.1).

Figure 5.56. System Log.

Each line in the log contains: a time stamp, 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. On Windows systems you can see the log by opening the file `[install_dir]otrs\var\log\otrs.log` with a text editor.

5.19. SQL queries via the SQL box

The "SQL Box" link in the admin area, opens a screen that lets you query the content of the tables in the OTRS database (see Figure 5.57 below). It is not possible to change the content of the tables, only queries are allowed.

The screenshot shows the OTRS Admin interface. At the top, there is a navigation bar with tabs: DASHBOARD, TICKETS, STATISTICS, CUSTOMERS, and ADMIN (which is highlighted). A search icon is also present. Below the navigation bar, the 'SQL Box' section is visible. It contains a 'Note' box stating: 'Here you can enter SQL to send it directly to the application database.' To the right of the note is the 'Options' section, which includes a text input for 'SQL' containing 'SELECT * FROM', a 'Limit' input set to '40', and a 'Result format' dropdown set to 'HTML'. A 'Run Query' button is located at the bottom of the 'Options' section. At the bottom of the page, it says 'Powered by OTRS 3.0.x CVS' and 'Top of page' with a link icon.

Figure 5.57. SQL Box.

5.20. Package manager

Using the "Package Manager" in the admin area, you can install and manage packages that extend the functionality of OTRS (see Figure 5.58 below). See the Additional applications section for a discussion on the extensions that are available from the OTRS repositories.

The screenshot shows the OTRS Admin interface with the 'Package Manager' section active. The navigation bar is the same as in Figure 5.57. The 'Package Manager' section has two main parts: 'Actions' and 'Online Repository'. The 'Actions' section includes a 'Browse...' button, an 'Install Package' button, a dropdown menu for the repository (currently set to '[-Master] http://ftp.otrs.org/'), and an 'Update repository information' button. The 'Online Repository' section displays a table with the following data:

NAME	VERSION	VENDOR	DESCRIPTION	ACTION
iPhoneHandle	0.9.4	OTRS AG	The iPhoneHandle Package.	Install

Below the 'Online Repository' table is a 'Local Repository' section, which also has a table with the same headers, but it contains the text 'No data found.' At the bottom of the page, it says 'Powered by OTRS 3.0.x CVS' and 'Top of page' with a link icon.

Figure 5.58. Package Manager.

In the package manager you can review the packages you currently have installed together with their version numbers.

You can install packages from a remote host by selecting the repository in the *Online Repository* section and clicking the *Update repository information* button. The available packages are displayed in the corresponding table. The right side of the screen shows the available packages. To install one, click on *Install*. After installation the package is displayed in the *Local Repository* section.

To upgrade an installed package, the list of available packages in the online repository will show *Upgrade* in the Action column for any package that has a higher version than the locally installed. Just click Upgrade and it will install the new package on your system.

In some cases, such as when your OTRS system is not connected to the Internet, you can also install packages you have downloaded to a local disk. Click the *Browse* button on the Actions side bar and select the .opm file on your disk. Click *Open* and then *Install Package*. After installation the package is displayed in the *Local Repository* section. You can use the same steps for updating a package that is already installed.

In special cases, you might want to configure the package manager, e.g., to use a proxy or to use a local repository. Just take a look at the available options in SysConfig under Framework:Core::Package.

Chapter 6. Configuration of the system

6.1. The config files of OTRS

All configuration files of OTRS are stored in the directory `Kernel` and in its sub directories. There is no need to manually change any other file than `Kernel/Config.pm`, because the rest of the files will be changed when the system gets upgraded. Just copy the configuration parameters from the other files into `Kernel/Config.pm` and change them to your needs. This file will never be touched during the upgrade process, so your manual settings are safe.

The file `Kernel/Config/Defaults.pm` contains the parameters of the central OTRS framework. It defines all basic system settings, like the mail configuration, database connection, default charset and standard language. The file `Kernel/Config/Files/Ticket.pm` contains all configuration parameters for the trouble ticket system.

In the directory `Kernel/Config/Files` there are some other files that are parsed when the OTRS login page is accessed. If additional applications like the FAQ or the file manager are installed, the configuration files for those can also be found in the mentioned path.

To keep compatibility with older OTRS versions, there is always a `.pm`- and a `.xml` file for every application and for the central framework. The `.pm` files are needed to parse all default and customized settings. The XML files are used by the graphical configuration frontend, which is available with OTRS 2.0 onwards, and which lets you set most of the configuration parameters of the system through your web browser. In future versions of OTRS the `.pm` files will be removed and the system will be configurable only via the `.xml` files. All parameters in `Kernel/Config/Defaults.pm` will be moved to `Kernel/Config/Files/Framework.xml` and all parameters for the trouble ticket system will be no longer stored in `Kernel/Config/Files/Ticket.pm` but in `Kernel/Config/Files/Ticket.xml`. The format of the configuration parameters will completely change to xml.

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

6.2. Configuring the system through the web interface

With OTRS 2.0 and subsequent versions, nearly all configuration parameters of the central framework or

additional 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 (see Figure 6.1 below).

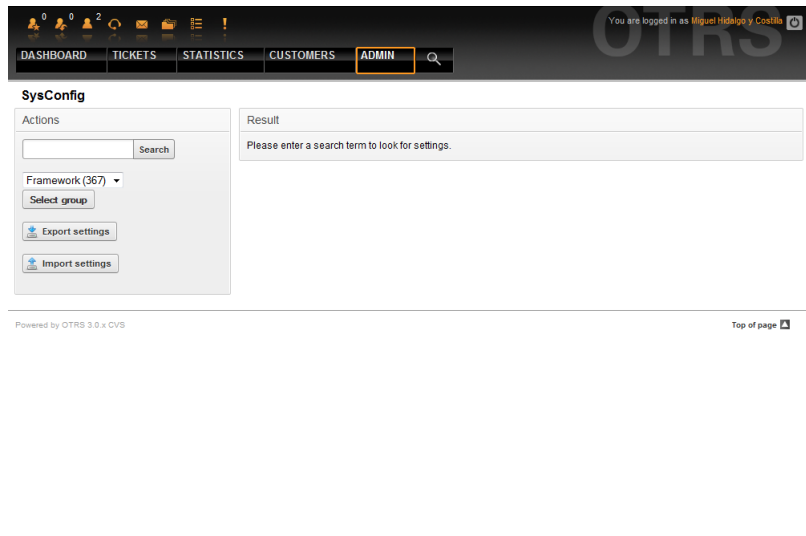


Figure 6.1. The graphical configuration frontend.

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

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 can be accessed if the application is selected from the groups listbox and the "Select group" button is pressed.

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

If you want to save all the changes you made to the configuration of your system, for example, to setup a

new installed system quickly, you can use the "Export settings" button, which will create a .pm file. To restore your own settings, just press the "Import settings" and select the .pm created before.

Note: For security reasons, the configuration parameters for the database connection cannot be changed in the SysConfig. 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 is to use Sendmail and should work out of the box.

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

7.1.2. Via SMTP server or smarthost

OTRS can send emails via SMTP (Simple Mail Transfer Protocol / RFC 821 (<http://www.ietf.org/rfc/rfc821.txt>)) or Secure SMTP. You will want to use this on non-UNIX platforms (e.g. Windows).

The SMTP server settings can be configured via the SysConfig (Framework::Core::Sendmail). If you don't see SMTPS available as an option, you miss the required Perl modules. In that case, please refer to "Installation of Perl modules required for OTRS" for instructions.

7.2. Receiving emails

7.2.1. Mail accounts configured via the OTRS GUI

OTRS is able to receive emails from POP3, POP3S, IMAP, and IMAPS mail accounts.

Configure your mail accounts via the PostMaster Mail Accounts link in the admin interface.

If a new mail account is created (see Figure 7.1 below), then its mail server name, login name and password must be specified. Also, you need to select the mail server type, which can be POP3, POP3S, IMAP or IMAPS. If you don't see server type you'd want to use available as an option, you miss the required Perl modules. In that case, please refer to "Installation of Perl modules required for OTRS" for instructions.

The screenshot shows the OTRS Admin interface. At the top, there is a navigation bar with tabs: DASHBOARD, TICKETS, STATISTICS, CUSTOMERS, and ADMIN (which is highlighted). Below the navigation bar, the page title is 'Mail Account Management'. On the left side, there is a sidebar with a 'Note' section containing text about email dispatching and a 'Go to overview' button. The main content area is titled 'Add Mail Account' and contains a form with the following fields:

- Type: POP3 (dropdown)
- Username: miguel.hidalgo (text input)
- Password: (password input)
- Host: mail.independence.com (text input)
- Example: mail.example.com (text input)
- Trusted: Yes (dropdown)
- Dispatching: Dispatching by email To: field (dropdown)
- Queue: Postmaster (dropdown)
- Valid: valid (dropdown)
- Comment: Example of mail account (text input)

At the bottom of the form, there are two buttons: 'Submit' and 'Cancel'. The footer of the page indicates 'Powered by OTRS 3.0.x CVS' and 'Top of page'.

Figure 7.1. Adding a mail account.

If you select "Yes" for "Trusted", any X-OTRS headers attached to an incoming message are evaluated and executed. Because the X-OTRS header can execute some actions in the ticket system you should only set "Trusted" to "Yes" for known senders. X-OTRS-Headers are used by the filter module in OTRS. The X-OTRS headers are explained in more detail. Any postmaster filter rules you'd have created are executed even if "Trusted" is set to "Yes".

The distribution of incoming messages can be controlled if they need to be sorted by queue or by the content of the To: field. If "Dispatching by selected queue" is selected for "Dispatching", all incoming messages will be sorted into the specified queue. The address where the mail was sent to is disregarded in this case. 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. You can link an address in the E-mail address management section of the admin area. If the address in the To field is linked with a queue, the new message will be sorted into the linked queue. If no link is found between the address in the To: field and queue then the message is sorted into the "Raw" queue in the system, which is the PostmasterDefaultQueue after a default installation.

All data for the mail accounts are saved in the OTRS database. The `otrs.PostMasterMailbox.pl` script, which is located in the `bin` directory of your OTRS installation, uses the settings in the database and fetches the mail. You can execute `./bin/otrs.PostMasterMailbox.pl` manually to check if all your mail settings are working properly.

On a normal installation, the mail will be fetched every 10 minutes by the `postmaster_mailbox` cron job. For further information about modifying cron jobs, please refer to the `cron jobs` chapter.

Note: When fetching mail OTRS deletes the mail from the POP or IMAP server. There's no option to also keep a copy on the server. If you'd want to achieve this, you'd probably best create forwarding rules on your mail server. Please consult your mail server documentation for details.

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

If you can't use mail accounts to get the email into OTRS, the command line program `bin/otrs.PostMaster.pl` might be a solution. It takes the mails via STDIN and pipes them directly into OTRS. That means email will be available in your OTRS system if the MDA (mail delivery agent, e.g. procmail) executes this program.

To test `bin/otrs.PostMaster.pl` without MDA, execute the command of the Script 7.1.

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

Script 7.1. Testing PostMaster without MDA.

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

Procmail is a very common e-mail filter in Linux environments. 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` and add the lines of the Script 7.2.

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

Script 7.2. Configuring procmail for OTRS.

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

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

In order to get email from your mail server, via a POP3 or IMAP mailbox, to the OTRS machine/local OTRS account and to procmail, use fetchmail (<http://fetchmail.berlios.de/>).

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 (see the Example 7-1 below).

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 7-1 above, all email will be forwarded to the local OTRS account, if the command **fetchmail -a** is executed. Set up a cronjob with this command if you want to fetch the mails regularly.

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

If you use the `bin/otrs.PostMaster.pl` or `bin/otrs.PostMasterMailbox.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 or change the customer ID, for example. More information about the X-OTRS headers are available in the chapter about adding mail accounts in the admin area of OTRS.

There are some default filter modules:

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

`Kernel::System::PostMaster::Filter::Match` is a default module to match on some email header (e.g. From, To, Subject, ...). It can set new email headers (e.g. X-OTRS-Ignore: yes or X-OTRS-Queue: spam) if a filter rule matches. The jobs of the Example 7-2 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 Example 7-3 can be used in `Kernel/Config.pm`

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

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

```
        'X-OTRS-Ignore' => 'yes',  
    },  
};
```

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

Chapter 8. Time related functions

8.1. Setting up business hours, holidays and time zones

Some functions in OTRS, like escalations and automatic unlocking of tickets, depend on a proper configuration of business hours, time zones and holidays. You can define these via the SysConfig interface, in Framework > Core::Time. You can also specify different sets of business hours, holidays and time zones as separate 'Calendars' in Framework > Core::Time::Calendar1, until Framework > Core::Time::Calendar9. Calendars can be defined on queue or on SLA levels. This means that, for example, you can specify a calendar with 5 x 8 business hours for your 'standard' SLA, but create a separate calendar with 7 x 24 support for your 'gold' SLA; and to have a calendar for your 'Support-USA' queue with a different time window than your 'Support-Japan' queue. OTRS can handle up to 99 different calendars.

8.1.1. Business Hours

Set up the working hours for your system in SysConfig Framework> Core::Time::TimeWorkingHours or for your specific calendar, in the calendar's configuration. OTRS can handle a granularity of one hour. Checking the marks in the boxes 8, 9, 10 ... 17 corresponds with business hours of 8 AM - 6 PM.

Only during the business hours: tickets can escalate, notifications for escalated and pending tickets will be sent and tickets will be unlocked.

8.1.2. Fixed date holidays

Holidays that are on a fixed date every year, such as New Year's Day and Fourth of July, can be specified in TimeVacationDays or in the corresponding section for the calendars 1-9.

Tickets will not escalate nor get unlocked on dates defined in TimeVacationDays.

Note: By default, OTRS ships with the *German* holidays installed.

8.1.3. TimeVacationDaysOneTime

Holidays that do not have a yearly fixed date, such as Easter, can be specified in TimeVacationDaysOneTime.

Tickets will not escalate and will not be unlocked on dates defined in `TimeVacationDaysOneTime`.

Note: OTRS does not ship with any One-Time holidays pre-installed. This means that you need to add holidays, such as Eastern or Thanksgiving, to the system when configuring OTRS.

8.2. Automated Unlocking

Locked tickets can be automatically unlocked by the system. This feature might be useful if, for example, an agent has locked tickets that need to be processed, but he can't work on them for some reason, like being on vacation. The automated unlock feature unlocks tickets after a given time, to ensure that no locked tickets will be forgotten and other agents can process them.

The amount of time before a ticket is unlocked can be specified in the settings for every queue. The module `bin/otrs.UnlockTickets.pl`, which is executed periodically as a cron job, performs the automated unlocking of tickets.

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

Tickets will be unlocked if all of the following conditions are met:

- There is an *unlock timeout* defined at the queue the ticket is in.
- The ticket is set to *locked*
- The ticket state is *open*.

The unlock timer will be reset if an agent adds a new external article to the ticket, It can be of any of the following types: *email-external*, *phone*, *fax*, *sms*, or *note-external*.

Also, if the last article in the ticket is created by an agent, and a customer adds another one, either via web or email response, the unlock timer will be reset.

The last event that will reset the unlock timer is when the ticket is assigned to another agent.

Chapter 9. Ticket responsibility and ticket watching

From OTRS 2.1 on, it is possible to define a person responsible for a ticket, additionally to its owner. Moreover, all activities connected with the ticket can be watched. These two functionalities are implemented with the TicketResponsible and TicketWatcher features, and facilitate the assignment of tasks and working within hierarchic team structures.

9.1. Ticket responsibility

The ticket responsible feature facilitates the complete processing of a ticket by an agent, other than the ticket owner. Thus an agent who has locked a ticket can pass it on to another agent, who is not the ticket owner, in order for the second to respond to a customer request. After the request has been dealt with, the first agent can withdraw the ticket responsibility from the second agent.

With the configuration parameter `Ticket::Responsible`, the ticket responsible feature can be activated. This will cause that 3 new icons appear in your toolbar.

Ticket responsibility can be assigned by calling up the ticket content and clicking on the "Responsible" link in the actions bar of a zoomed ticket in the agent interface (see the Figure 9.1 below).

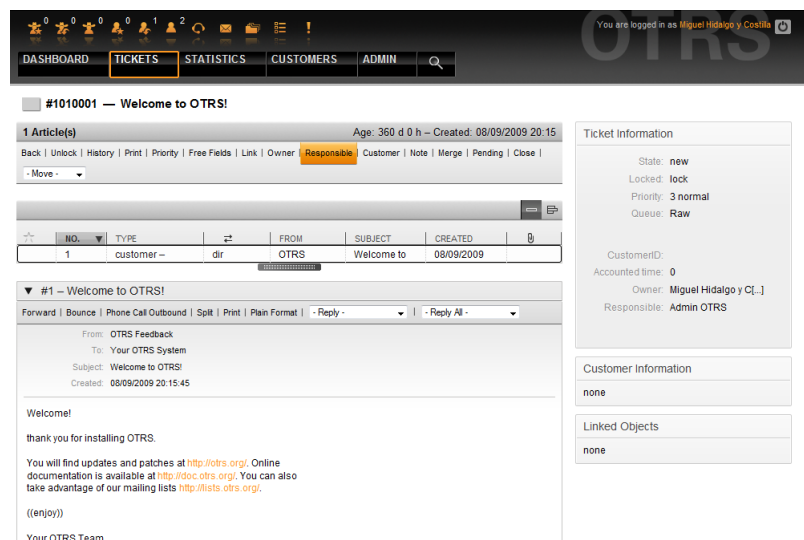


Figure 9.1. Changing the responsible of a ticket in its zoomed view.

After clicking on "Responsible", a pop-up to change the responsible of that ticket will open (see Figure 9.2 below). This template can also be used to send a message to the new responsible agent.

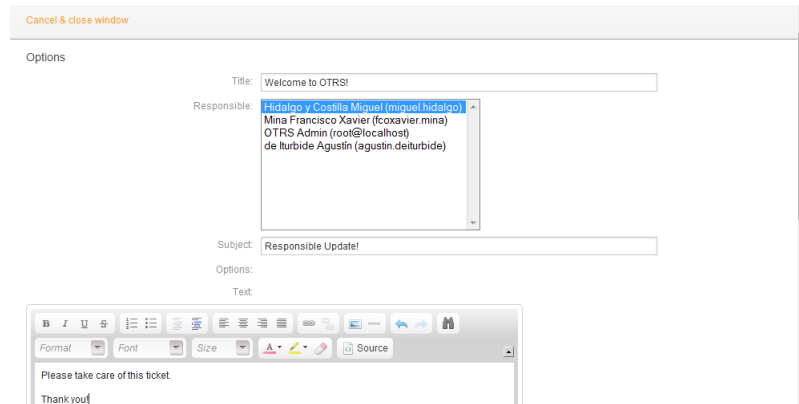


Figure 9.2. Change a ticket's responsible pop-up.

The list of all tickets for which an agent is responsible from, can be accessed through the Responsible View, which is shown as 3 icons in the toolbar of the OTRS agent interface, as soon as the ticket responsible feature gets activated.

9.2. Ticket watching

From OTRS 2.1 on, agents (e.g. heads of department) can watch certain tickets within the system without processing them, by using the TicketWatcher feature.

The TicketWatcher feature can be activated with the configuration parameter `Ticket::Watcher`, which will add 3 new icons to your toolbar. Using `Ticket::WatcherGroup` one or more user groups with permission to watch tickets can be defined.

In order to watch a ticket, go to its zoomed view and click on the "Subscribe" link in the ticket activities menu (see Figure 9.3 below).

The screenshot shows the OTRS web interface. At the top, there's a navigation bar with tabs: DASHBOARD, TICKETS (selected), STATISTICS, CUSTOMERS, and ADMIN. A user is logged in as Miguel Hidalgo y Castiella. The main content area displays ticket #1010001 titled "Welcome to OTRS!". It shows 2 articles, with the first article being the ticket itself. The ticket details on the right indicate it's a new, locked ticket with a priority of 3 normal and a queue of Raw. The customer information is none. The ticket body contains a welcome message and links to OTRS documentation and mailing lists. The "Subscribe" button is highlighted in the ticket activities menu.

Figure 9.3. Subscribing to a ticket in its zoomed view.

If you no longer want to watch a specific ticket, go to its zoomed view and click on the "Unsubscribe" link in the ticket activities menu (see Figure 9.4 below).

This screenshot is similar to Figure 9.3, showing the same ticket #1010001. However, in the ticket activities menu, the "Unsubscribe" button is now highlighted, indicating the user is about to unsubscribe from the ticket. The rest of the interface, including the ticket details and body, remains the same.

Figure 9.4. Unsubscribing from a ticket in its zoomed view.

The list of all watched tickets can be accessed through the Watched View (see Figure 9.5 below), which is shown as 3 icons in the toolbar of the OTRS agent interface, as soon as the ticket watcher feature gets activated.

You are logged in as Miguel Hidalgo y Costilla

DASHBOARD TICKETS STATISTICS CUSTOMERS ADMIN

My Watched Tickets: All

All 1 New Article 0 Pending 0 Reminder Reached 0

Bulk 1-1 of 1 - Page: 1 2 3 M L

TICKET#	AGE	FROM / SUBJECT	STATE	LOCKED	QUEUE	OWNER	CUSTOMERID
1010001	367 d 6 h	OTRS Feedback Welcome to OTRS!	new	lock	Raw	Miguel Hidalgo y Costi[...]	

Powered by OTRS 3.0.x CVS Top of page

Figure 9.5. Watched tickets view.

Chapter 10. Customize the PDF output

This section handles the configurable options for PDF output in OTRS.

If you would use the Print action from anywhere within the OTRS interface, this would generate a formatted PDF file. You can deactivate this by modifying the configuration parameter `PDF` . In that case, HTML output will be used instead.

You can adjust the look of the files generated by OTRS by creating your own logo and add it to `PDF::LogoFile` . You can use `PDF::PageSize` to define the standard page size of the generated pdf file (DIN-A4 or Letter) and `PDF::MaxPages` specifies the maximum number of pages for a pdf file, which is useful if a user tries to generate a very big output file by accident.

The Perl CPAN modules `PDF::API2` and `Compress::Zlib`, must be installed for the generation of pdf files. In many distributions they are available as packages and can be easily installed, using the respective package manager. In case this is not possible, they have to be installed with CPAN. For further information about installing Perl modules, please refer to the "Installation of Perl modules" chapter.

Chapter 11. Using external backends

11.1. Customer data

OTRS is able to handle different customer data attributes, like username, email address, phone number, etc. This information will be displayed in both, the agent and the customer frontends. These attributes are also used for the authentication of customers.

The used / displayed customer data are completely customizable, but without the following data OTRS won't work, because this information is always needed for the authentication of customers:

- User login
- Email address
- Customer ID

Use configuration parameters of the Script 11.1 in your `Kernel/Config.pm` file, if you want to display customer information in your agent interface.

```
# Ticket::Frontend::CustomerInfo*
# (show customer 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;
```

Script 11.1. Kernel/Config.pm configuration parameters.

11.2. Customer user backend

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

11.2.1. Database (Default)

The Example 11-1 shows the configuration of a DB customer backend, which uses customer data stored in the OTRS database.

Example 11-1. Configuring a DB customer backend

```
# CustomerUser (customer database backend and settings)
$Self->{CustomerUser} = {
    Name => 'Database Datasource',
    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 unique 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 not own tickets in customer panel, CompanyTickets
    # CustomerUserExcludePrimaryCustomerID => 0,
    # # generate auto logins
    # AutoLoginCreation => 0,
    # AutoLoginCreationPrefix => 'auto',
    # # admin can change customer preferences
    # AdminSetPreferences => 1,
    # # cache time to live in sec. - cache any database queries
    # CacheTTL => 0,
    # # 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-lin
        [ 'UserTitle', 'Title', 'title', 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, 0, 'var', "", 0 ],
[ 'UserEmail', 'Email', 'email', 1, 1, 'var', "", 0 ],

# [ 'UserEmail', 'Email', 'email', 1, 1, 'var', '$Env{"CGIHandle"}?Ac
[ 'UserCustomerID', 'CustomerID', 'customer_id', 0, 1, 'var', "", 0 ],

# [ 'UserCustomerIDs', 'CustomerIDs', 'customer_ids', 1, 0, 'var', "", 0 ],
[ 'UserPhone', 'Phone', 'phone', 1, 0, 'var', "", 0 ],
[ 'UserFax', 'Fax', 'fax', 1, 0, 'var', "", 0 ],
[ 'UserMobile', 'Mobile', 'mobile', 1, 0, 'var', "", 0 ],
[ 'UserStreet', 'Street', 'street', 1, 0, 'var', "", 0 ],
[ 'UserZip', 'Zip', 'zip', 1, 0, 'var', "", 0 ],
[ 'UserCity', 'City', 'city', 1, 0, 'var', "", 0 ],
[ 'UserCountry', 'Country', 'country', 1, 0, 'var', "", 0 ],
[ 'UserComment', 'Comment', 'comments', 1, 0, 'var', "", 0 ],
[ 'ValidID', 'Valid', 'valid_id', 0, 1, 'int', "", 0 ],
],
# default selections
Selections => {
    UserTitle => {
        'Mr.' => 'Mr.',
        'Mrs.' => 'Mrs.',
    },
},
};

```

If you want to customize the customer data, change the column's headers or add new ones to the `customer_user` table in the OTRS database. The Script 11.2 shows how to add a new field for room number.

```

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

```

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

```

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

```

```

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

```

```

mysql> quit
Bye
linux:~#

```


Script 11.2. Adding a room field to the customer_user table.

Now add the new column to the MAP array in `Kernel/Config.pm`, as shown in the Script 11.3.

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

Script 11.3. Adding a room field to the Kernel/Config.pm file.

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

11.2.1.1. Customer with multiple IDs (Company tickets)

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

should be accessible for a customer. To use company tickets, a new column, with the IDs that should be accessible for a customer, has to be added to the `customer_user` table in the OTRS database (see Script 11.4 below).

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

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

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

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

```
mysql> quit
Bye
linux:~#
```

Script 11.4. Adding a customer id field to the customer_user table.

Now the new column has to be added to the MAP array in `Kernel/Config.pm`, as shown in the Script 11.5.

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

Script 11.5. Adding a customer id field to the Kernel/Config.pm file.

Now, the new column for the multi customer IDs can be edited via the agent interface, in the section for the customer management.

To ensure that one customer can access the tickets of other customers, add the IDs of this other users into the new field for the multiple customer IDs. Each ID has to be separated by a semicolon (see Example 11-2 below).

Example 11-2. Using company tickets with a DB backend

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

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

11.2.2. LDAP

If you have a LDAP directory with your customer data, you can use it as customer backend with OTRS, as shown in the Example 11-3.

Example 11-3. Configuring a LDAP customer backend

```

# CustomerUser
# (customer ldap backend and settings)
$Self->{CustomerUser} = {
    Name => 'LDAP Data Source',
    Module => 'Kernel::System::CustomerUser::LDAP',
    Params => {
        # ldap host
        Host => 'bay.csuhayward.edu',
        # ldap base dn
        BaseDN => 'ou=seas,o=csuh',
        # search scope (one|sub)
        SSCOPE => 'sub',
        # The following is valid but would only be necessary if the
        # anonymous user does NOT have permission to read from the LDAP tree
        UserDN => "",
        UserPw => "",
        # in case you want to add always one filter to each ldap query, use
        # this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter => '(objectclass=us
        AlwaysFilter => "",
        # if your frontend is e. g. iso-8859-1 and the charset of your
        # ldap server is utf-8, use these options.
        SourceCharset => 'utf-8',
        DestCharset => 'iso-8859-1',
        # if both your frontend and your LDAP are unicode, use this:
        SourceCharset => 'utf-8',
        DestCharset => 'utf-8',
        # Net::LDAP new params (if needed - for more info see perldoc Net::LDAP)
        Params => {
            port => 389,
            timeout => 120,
            async => 0,
            version => 3,
        },
    },
    # customer unique id
    CustomerKey => 'uid',
    # customer #
    CustomerID => 'mail',
    CustomerUserListFields => ['cn', 'mail'],
    CustomerUserSearchFields => ['uid', 'cn', 'mail'],
    CustomerUserSearchPrefix => "",
    CustomerUserSearchSuffix => '*',
    CustomerUserSearchListLimit => 250,
    CustomerUserPostMasterSearchFields => ['mail'],
    CustomerUserNameFields => ['givenname', 'sn'],
    # show not own tickets in customer panel, CompanyTickets
    CustomerUserExcludePrimaryCustomerID => 0,
    # add an ldap filter for valid users (expert setting)
    CustomerUserValidFilter => '(! (description=locked))',

```

```

# administrator can't change customer preferences
AdminSetPreferences => 0,
# # cache time to live in sec. - cache any database queries
# CacheTTL => 0,
Map => [
  # note: Login, Email and CustomerID are mandatory!
  # var, frontend, storage, shown (1=always,2=lite), required, storage-type, http-link
  [ 'UserTitle',      'Title',      'title',      1, 0, 'var', "", 0 ],
  [ 'UserFirstname',  'Firstname',  'givenname', 1, 1, 'var', "", 0 ],
  [ 'UserLastname',   'Lastname',   'sn',        1, 1, 'var', "", 0 ],
  [ 'UserLogin',      'Username',   'uid',       1, 1, 'var', "", 0 ],
  [ 'UserEmail',      'Email',      'mail',      1, 1, 'var', "", 0 ],
  [ 'UserCustomerID', 'CustomerID', 'mail',      0, 1, 'var', "", 0 ],
#   [ 'UserCustomerIDs', 'CustomerIDs', 'second_customer_ids', 1, 0, 'var', "", 0 ],
  [ 'UserPhone',      'Phone',      'telephonenumber', 1, 0, 'var', "", 0 ],
  [ 'UserAddress',    'Address',    'postaladdress', 1, 0, 'var', "", 0 ],
  [ 'UserComment',    'Comment',    'description', 1, 0, 'var', "", 0 ],
],
};

```

If additional customer attributes are stored in your LDAP directory, such as a manager's name, a mobile phone number, or a department, and if you want to display this information in OTRS, just expand the MAP array in `Kernel/Config.pm` with the entries for these attributes, as shown in the Script 11.6.

```

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

```

Script 11.6. Adding new fields to the Kernel/Config.pm file.

11.2.2.1. Customer with multiple IDs (Company tickets)

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

To use company tickets a new field has to be added to the LDAP directory that contains the IDs that should be accessible for a customer.

If the new field in the LDAP directory has been created the new entry has to be added to the MAP array in `Kernel/Config.pm`, as shown in the Script 11.7.

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

Script 11.7. Mapping new fields to the Kernel/Config.pm file.

The field for the multiple customer IDs has to be edited directly in the LDAP directory, OTRS can only read from LDAP, not write to it.

To ensure the access for a customer to the tickets of other customers, add the customer IDs of the customers that tickets should be accessed to the new field in your LDAP directory. Each ID has to be separated by a semicolon (see the Example 11-4 below).

Example 11-4. Using Company tickets with a LDAP backend

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

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

11.2.3. Use more than one customer backend with OTRS

If you want to utilize more than one customer data source used with OTRS (e.g. an LDAP and a database backend), the `CustomerUser` config parameter should be expanded with a number, e.g. "CustomerUser1", "CustomerUser2" (see the Example 11-5 below).

Example 11-5. Using more than one customer backend with OTRS

The following configuration example shows usage of both an LDAP and a database customer backend with OTRS.

```
# 1. Customer user backend: DB
# (customer database backend and settings)
$Self->{CustomerUser1} = {
    Name => 'Customer Database',
    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 unique 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 not own tickets in customer panel, CompanyTickets
# CustomerUserExcludePrimaryCustomerID => 0,
# # generate auto logins
# AutoLoginCreation => 0,
# AutoLoginCreationPrefix => 'auto',
# # admin can change customer preferences
# AdminSetPreferences => 1,
# # cache time to live in sec. - cache any database queries
# CacheTTL => 0,
# # 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-lin
    [ 'UserTitle',      'Title',      'title',      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, 0, 'var', "", 0 ],
    [ 'UserEmail',     'Email',    'email',      1, 1, 'var', "", 0 ],
    [ 'UserCustomerID', 'CustomerID', 'customer_id', 0, 1, 'var', "", 0 ],
    [ 'UserPhone',     'Phone',    'phone',      1, 0, 'var', "", 0 ],
    [ 'UserFax',       'Fax',      'fax',        1, 0, 'var', "", 0 ],
    [ 'UserMobile',    'Mobile',   'mobile',     1, 0, 'var', "", 0 ],
    [ 'UserStreet',    'Street',   'street',     1, 0, 'var', "", 0 ],
    [ 'UserZip',       'Zip',      'zip',        1, 0, 'var', "", 0 ],
    [ 'UserCity',      'City',     'city',       1, 0, 'var', "", 0 ],
    [ 'UserCountry',   'Country',  'country',    1, 0, 'var', "", 0 ],
    [ 'UserComment',   'Comment',  'comments',   1, 0, 'var', "", 0 ],
    [ 'ValidID',       'Valid',    'valid_id',   0, 1, 'int', "", 0 ],

```

```

],
# default selections
Selections => {
    UserTitle => {
        'Mr.' => 'Mr.',
        'Mrs.' => 'Mrs.',
    },
},
};

# 2. Customer user backend: LDAP
# (customer ldap backend and settings)
$Self->{CustomerUser2} = {
    Name => 'LDAP Datasource',
    Module => 'Kernel::System::CustomerUser::LDAP',
    Params => {
        # ldap host
        Host => 'bay.csuhayward.edu',
        # ldap base dn
        BaseDN => 'ou=seas,o=csuh',
        # search scope (one|sub)
        SSCOPE => 'sub',
        # The following is valid but would only be necessary if the
        # anonymous user does NOT have permission to read from the LDAP tree
        UserDN => "",
        UserPw => "",
        # in case you want to add always one filter to each ldap query, use
        # this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter => '(objectclass=us
        AlwaysFilter => "",
        # if both your frontend and your LDAP are unicode, use this:
        # SourceCharset => 'utf-8',
        # DestCharset    => 'utf-8',
        # if your frontend is e. g. iso-8859-1 and the character set of your
        # ldap server is utf-8, use these options:
        # 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 unique id
    CustomerKey => 'uid',
    # customer #
    CustomerID => 'mail',
    CustomerUserListFields => ['cn', 'mail'],
    CustomerUserSearchFields => ['uid', 'cn', 'mail'],
    CustomerUserSearchPrefix => "",
    CustomerUserSearchSuffix => '*',

```

```

CustomerUserSearchListLimit => 250,
CustomerUserPostMasterSearchFields => ['mail'],
CustomerUserNameFields => ['givenname', 'sn'],
# show not own tickets in customer panel, CompanyTickets
CustomerUserExcludePrimaryCustomerID => 0,
# add a ldap filter for valid users (expert setting)
# CustomerUserValidFilter => '(!(description=locked))',
# 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-lin
    [ 'UserTitle',      'Title',      'title',          1, 0, 'var', "", 0 ],
    [ 'UserFirstname', 'Firstname',  'givenname',      1, 1, 'var', "", 0 ],
    [ 'UserLastname',  'Lastname',  'sn',            1, 1, 'var', "", 0 ],
    [ 'UserLogin',     'Username',  'uid',           1, 1, 'var', "", 0 ],
    [ 'UserEmail',     'Email',     'mail',          1, 1, 'var', "", 0 ],
    [ 'UserCustomerID', 'CustomerID', 'mail',          0, 1, 'var', "", 0 ],
#    [ 'UserCustomerIDs', 'CustomerIDs', 'second_customer_ids', 1, 0, 'var', "", 0 ],
    [ 'UserPhone',     'Phone',     'telephonenumber', 1, 0, 'var', "", 0 ],
    [ 'UserAddress',   'Address',   'postaladdress',  1, 0, 'var', "", 0 ],
    [ 'UserComment',   'Comment',   'description',    1, 0, 'var', "", 0 ],
],
};

```

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

11.3. Backends to authenticate agents and customers

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

11.3.1. Authentication backends for agents

11.3.1.1. DB (Default)

The backend to authenticate agents which is used per default is the OTRS database. Agents can be added and edited via the user management interface in the admin area (see Example 11-6 below).

Example 11-6. Authenticate agents against a DB backend

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

11.3.1.2. LDAP

If a LDAP directory has all your agent data stored, you can use the LDAP module to authenticate your users in OTRS (see Example 11-7 below). This module has only read access to the LDAP tree, that means you can't edit your users via the user management interface .

Example 11-7. Authenticate agents against a LDAP backend

```
# This is an example configuration for an LDAP auth. backend.
# (Make sure Net::LDAP is installed!)
$Self->{'AuthModule'} = 'Kernel::System::Auth::LDAP';
$Self->{'AuthModule::LDAP::Host'} = 'ldap.example.com';
$Self->{'AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';
$Self->{'AuthModule::LDAP::UID'} = 'uid';

# Check if the user is allowed to auth in a posixGroup
# (e. g. user needs to be in a group xyz to use otrs)
$Self->{'AuthModule::LDAP::GroupDN'} = 'cn=otrsallow,ou=posixGroups,dc=example,dc=com';
$Self->{'AuthModule::LDAP::AccessAttr'} = 'memberUid';
# for ldap posixGroups objectclass (just uid)
# $Self->{'AuthModule::LDAP::UserAttr'} = 'UID';
# for non ldap posixGroups objectclass (with full user dn)
# $Self->{'AuthModule::LDAP::UserAttr'} = 'DN';

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

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

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

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

```

    async => 0,
    version => 3,
};

```

The configuration parameters shown in the Script 11.8 can be used to synchronize the user data from your LDAP directory into your local OTRS database. This reduces the number of requests to your LDAP server and speeds up the authentication with OTRS. The data sync is done when the agent authenticates the first time. Although the data can be synced into the local OTRS database, the LDAP directory is the last instance for the authentication, so an inactive user in the LDAP tree can't authenticate to OTRS, even when the account data is already stored in the OTRS database. The agent data in the LDAP directory can't be edited via the web interface of OTRS, so the data has to be managed directly in the LDAP tree.

```

# agent data sync against ldap
$Self->{'AuthSyncModule'} = 'Kernel::System::Auth::Sync::LDAP';
$Self->{'AuthSyncModule::LDAP::Host'} = 'ldap://ldap.example.com/';
$Self->{'AuthSyncModule::LDAP::BaseDN'} = 'dc=otrs, dc=org';
$Self->{'AuthSyncModule::LDAP::UID'} = 'uid';
$Self->{'AuthSyncModule::LDAP::SearchUserDN'} = 'uid=sys, ou=user, dc=otrs, dc=org';
$Self->{'AuthSyncModule::LDAP::SearchUserPw'} = 'some_pass';
$Self->{'AuthSyncModule::LDAP::UserSyncMap'} = {
    # DB -> LDAP
    UserFirstname => 'givenName',
    UserLastname  => 'sn',
    UserEmail     => 'mail',
};
[...]

# AuthSyncModule::LDAP::UserSyncInitialGroups
# (sync following group with rw permission after initial create of first agent
# login)
$Self->{'AuthSyncModule::LDAP::UserSyncInitialGroups'} = [
    'users',
];

```

Script 11.8. Synchronizing the user data from the LDAP directory into the OTRS database.

11.3.1.3. HTTPBasicAuth for agents

If you want to implement a "single sign on" solution for all your agents, you can use HTTP basic authentication (for all your systems) and the HTTPBasicAuth module for OTRS (see Example 11-8 below).

Example 11-8. Authenticate agents using HTTPBasic

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

# Note:
#
# If you use this module, you should use as fallback
# the following configuration settings if the user is not authorized
# 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';
```

11.3.1.4. Radius

The configuration parameters shown in the Example 11-9 can be used to authenticate agents against a radius server.

Example 11-9. Authenticate agents against a radius backend

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

11.3.2. Authentication backends for customers**11.3.2.1. Database (Default)**

The default used authentication backend for customers in OTRS is the OTRS database. With this backend, all customer data can be edited via the web interface of OTRS (see the Example 11-10 below).

Example 11-10. Customer user authentication against a DB backend

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

```

$Self->{'Customer::AuthModule::DB::Table'} = 'customer_user';
$Self->{'Customer::AuthModule::DB::CustomerKey'} = 'login';
$Self->{'Customer::AuthModule::DB::CustomerPassword'} = 'pw';
##$Self->{'Customer::AuthModule::DB::DSN'} = "DBI:mysql:database=customerdb;host=customerdbh";
$Self->{'Customer::AuthModule::DB::User'} = "some_user";
$Self->{'Customer::AuthModule::DB::Password'} = "some_password";

```

11.3.2.2. LDAP

If you have a LDAP directory with all your customer data, you can use the LDAP module to authenticate your customers to OTRS (see Example 11-11 below). Because this module has only read-access to the LDAP backend, it is not possible to edit the customer data via the web interface of OTRS.

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

```

# This is an example configuration for an LDAP auth. backend.
# (make sure Net::LDAP is installed!)
$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::LDAP';
$Self->{'Customer::AuthModule::LDAP::Host'} = 'ldap.example.com';
$Self->{'Customer::AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';
$Self->{'Customer::AuthModule::LDAP::UID'} = 'uid';

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

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

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

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

# Net::LDAP new params (if needed - for more info see perldoc Net::LDAP)
$Self->{'Customer::AuthModule::LDAP::Params'} = {

```

```

port => 389,
timeout => 120,
async => 0,
version => 3,
};

```

11.3.2.3. HTTPBasicAuth for customers

If you want to implement a "single sign on" solution for all your customer users, you can use HTTPBasic authentication (for all your systems) and use the HTTPBasicAuth module with OTRS (no login is needed with OTRS any more). See the Example 11-12 below.

Example 11-12. Customer user authentication with HTTPBasic

```

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

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

```

11.3.2.4. Radius

The settings shown in the Example 11-13 can be used to authenticate your customers against a radius server.

Example 11-13. Customer user authentication against a radius backend

```

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

```

11.4. Customize the customer self registration

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

The following example shows how you can specify a required field in the customer database, in this case, to store the room number of a customer.

11.4.1. Customizing the web interface

To display the new field for the room number in the customer.pl web interface, the .dtl file, which is responsible for the layout in this interface, has to be modified. Edit the `Kernel/Output/HTML/Standard/CustomerLogin.dtl` file and add the new field around line 80 (see Script 11.9 below).

```
[...]
<div class="NewLine">
    <label for="Room">$Text{"Room{CustomerUser}"}</label>
    <input title="$Text{"Room Number"}" name="Room" type="text" id="UserRoom" maxlength="50" />
</div>
[...]
```

Script 11.9. Displaying a new field in the web interface.

11.4.2. Customer mapping

In the next step, the customer mapping has to be expanded with the new entry for the room number. To ensure that the changes are not lost after an update, put the "CustomerUser" settings from the `Kernel/Config/Defaults.pm` into the `Kernel/Config.pm`. Now change the MAP array and add the new room number field, as shown in the Script 11.10.

```
# CustomerUser
# (customer 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 unique id
    CustomerKey => 'login',
    # customer #
    CustomerID => 'customer_id',
    CustomerValid => 'valid_id',
    CustomerUserListFields => ['first_name', 'last_name', 'email'],
#   CustomerUserListFields => ['login', 'first_name', 'last_name', 'customer_id', 'email']
    CustomerUserSearchFields => ['login', 'last_name', 'customer_id'],
    CustomerUserSearchPrefix => "",
    CustomerUserSearchSuffix => '*',
    CustomerUserSearchListLimit => 250,
    CustomerUserPostMasterSearchFields => ['email'],
    CustomerUserNameFields => ['salutation', 'first_name', 'last_name'],
    CustomerUserEmailUniqCheck => 1,
#   # show not own tickets in customer panel, CompanyTickets
#   CustomerUserExcludePrimaryCustomerID => 0,
#   # generate auto logins
#   AutoLoginCreation => 0,
#   AutoLoginCreationPrefix => 'auto',
#   # admin can change customer preferences
#   AdminSetPreferences => 1,
#   # cache time to live in sec. - cache database queries
#   CacheTTL => 0,
#   # 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-lin
        [ 'UserTitle',      'Title',      'title',      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, 0, 'var', "", 0 ],
        [ 'UserEmail',     'Email',    'email',      1, 1, 'var', "", 0 ],
        [ 'UserCustomerID', 'CustomerID', 'customer_id', 0, 1, 'var', "", 0 ],
        [ 'UserPhone',     'Phone',    'phone',      1, 0, 'var', "", 0 ],
        [ 'UserFax',       'Fax',      'fax',        1, 0, 'var', "", 0 ],
        [ 'UserMobile',    'Mobile',   'mobile',     1, 0, 'var', "", 0 ],
        [ 'UserRoom',      'Room',     'room',       1, 0, 'var', "", 0 ],
        [ 'UserStreet',    'Street',   'street',     1, 0, 'var', "", 0 ],
        [ 'UserZip',       'Zip',      'zip',        1, 0, 'var', "", 0 ],
        [ 'UserCity',      'City',     'city',       1, 0, 'var', "", 0 ],
        [ 'UserCountry',   'Country',  'country',    1, 0, 'var', "", 0 ],
        [ 'UserComment',   'Comment',  'comments',   1, 0, 'var', "", 0 ],
        [ 'ValidID',       'Valid',    'valid_id',   0, 1, 'int', "", 0 ],
    ],
    # default selections

```

```

Selections => {
    UserTitle => {
        'Mr.' => 'Mr.',
        'Mrs.' => 'Mrs.',
    },
},
};

```

Script 11.10. Changing the map array.

11.4.3. Customize the customer_user table in the OTRS DB

The last step is to add the new room number column to the customer_user table in the OTRS database (see Script 11.11 below). In this column, the entries for the room numbers will be stored.

```

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

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

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

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

mysql> quit
Bye
linux:~#

```

Script 11.11. Adding a new column to the customer_user table.

Now you took all the steps involved, and the new field for the room should be displayed in the customer.pl panel. New customers should have to insert their room number if they register a new account. If you use apache and use mod_perl for OTRS, you should restart the web server to activate the changes.

Chapter 12. States

12.1. Predefined states

In OTRS, it is possible to change the predefined ticket states and their types, or even add new ones. Two attributes are important for a state: the name (state-name) and type (state-type).

The default states of OTRS are: 'closed successful', 'closed unsuccessful', 'merged', 'new', 'open', 'pending auto close+', 'pending auto close-', 'pending reminder' and 'removed'.

12.1.1. New

Tickets are usually only in this state when they are created from incoming e-mails.

12.1.2. Open

The default state for tickets assigned to queues and agents.

12.1.3. Pending reminder

After the pending time has expired, the bug owner will receive a reminder email concerning the ticket. If the ticket is not locked, the reminder will be sent to all agents in the queue. Reminder tickets will only be sent out within business hours, and are repeated every 24 hours until the state of the ticket is changed by the agent. Time spent by the ticket in this status will still add towards the escalation time calculations.

12.1.4. Pending auto close-

Tickets in this status will be set to Closed Unsuccessful if the pending time has expired. Time spent by the ticket in this status will still add towards the escalation time calculations.

12.1.5. Pending auto close+

Tickets in this status will be set to Closed Successful if the pending time has expired. Time spent by the ticket in this status will still add towards the escalation time calculations.

12.1.6. Merged

State for tickets that have been merged with other tickets.

12.1.7. Closed Successful

End state for tickets that have been successfully resolved. Depending on your configuration, you might not be able to reopen closed tickets.

12.1.8. Closed Unsuccessful

End state for tickets that have not been successfully resolved. Depending on your configuration, you might not be able to reopen closed tickets.

12.2. Customizing states

Every state has a name (state-name) and a type (state-type). You can freely choose the name of a state, via the admin area of OTRS, in the section for states. The state types can not be changed in the web interface, but the database has to be modified if you want to add new types or change existing names. We would generally recommend not to modify the default state types, as this can cause unexpected results. For instance, escalation calculations and the unlock feature are based on certain state types.

In the section for states, in the admin area of OTRS, the name of already existing states can be changed or new states can be added. If the state "new" has been changed via the web interface, this change also has to be configured via the config file `Kernel/Config.pm` or via the SysConfig interface. The settings specified in the Script 12.1 have to be modified to ensure that OTRS works with the changed state for "new".

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

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

Script 12.1. Modifying the Kernel/Config.pm settings.

If a new state type should be added, the ticket_state_type table in the OTRS database needs to be modified with a database client program, as shown in the Script 12.2.

```
linux:~# mysql -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 23 to server version: 5.0.16-Debian_1-log

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

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

Database changed
mysql> insert into ticket_state_type (name,comments) values ('own','Own
state type');
Query OK, 1 row affected (0.00 sec)

mysql> quit
Bye
linux:~#
```

Script 12.2. Modifying the OTRS database.

Now it is possible to use the new state type you just created in the section for states, in the admin area. After a state has been linked with this new state type, also the configuration of OTRS has to be changed, to ensure that the new state is used. Just modify the following options via SysConfig:

Ticket -> Frontend::Agent::Ticket::ViewPhoneNew > AgentTicketPhone###StateDefault - to define the default next state for new phone tickets.

Ticket -> Frontend::Agent::Ticket::ViewPhoneNew > AgentTicketPhone###StateType - to define the available next states for new phone tickets.

Ticket -> Frontend::Agent::Ticket::ViewEmailNew > AgentTicketEmail###StateDefault - to define the default next state for new email tickets.

Ticket -> Frontend::Agent::Ticket::ViewEmailNew > AgentTicketEmail###StateType - to define the available next states for new email tickets.

Ticket -> Frontend::Agent::Ticket::ViewPhoneOutbound > AgentTicketPhoneOutbound###State - to define the default next state for new phone articles.

Ticket -> Frontend::Agent::Ticket::ViewPhoneOutbound > AgentTicketPhoneOutbound###StateType - to define the available next states for new phone articles.

Ticket:Frontend::Agent::Ticket::ViewMove:Ticket::DefaultNextMoveStateType - to define the default next state after moving a ticket.

Ticket -> Frontend::Agent::Ticket::ViewBounce > StateDefault - to define the default next state after bouncing a ticket.

Ticket -> Frontend::Agent::Ticket::ViewBounce > StateType - to define the available next states in the bounce screen.

Ticket -> Frontend::Agent::Ticket::ViewBulk > StateDefault - to define the default next state in a bulk action.

Ticket -> Frontend::Agent::Ticket::ViewBulk > StateType - to define the available next states in the bulk action screen.

Ticket -> Frontend::Agent::Ticket::ViewClose > StateDefault - to define the default next state after closing a ticket.

Ticket -> Frontend::Agent::Ticket::ViewClose > StateType - to define the available next states in the close screen.

Ticket -> Frontend::Agent::Ticket::ViewCompose > StateDefault - to define the default next state in the Compose (reply) screen.

Ticket -> Frontend::Agent::Ticket::ViewCompose > StateType - to define the available next states in the Compose (reply) screen.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateDefault - to define the default next state after forwarding a ticket.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateType - to define the available next states in the Forward screen.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateDefault - to define the default next state of a ticket in the free text screen.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateType - to define the available next states in the free text screen.

Ticket -> Core::PostMaster > PostmasterDefaultState - to define the state of tickets created from emails.

Ticket -> Core::PostMaster > PostmasterFollowUpState - to define the state of tickets after a follow-up has been received.

Ticket -> Core::PostMaster > PostmasterFollowUpStateClosed - to define the state of tickets after a follow-up has been received and the ticket was already closed.

Ticket -> Core::Ticket > ViewableStateType - to define the state types that are displayed at various places in the system, for example in the Queueview.

Ticket -> Core::Ticket > UnlockStateType - to define the state types for unlocked tickets.

Ticket -> Core::Ticket > PendingReminderStateType - to define the state type for reminder tickets.

Ticket -> Core::Ticket > PendingAutoStateType - to define the state type for Pending Auto tickets.

Ticket -> Core::Ticket > StateAfterPending - to define the state a ticket is set to after the Pending Auto timer of the configured state has expired.

Chapter 13. Modifying ticket priorities

OTRS comes with five default priority levels, which can be modified via the "Priority" option in the Admin console. When creating a customized list of priorities, please keep in mind that they are sorted alphabetically in the priority selection box, in the user interface. Also, OTRS orders tickets by internal database IDs in the QueueView.

Note: To keep the consistency of the data for OTRS it is not possible to delete priorities, but only deactivate them. To do so, set them to *invalid* or *invalid-temporarily*.

Important: If a new priority was added or if a one was changed, you might also want to modify some values in SysConfig:

- Ticket:Core::Postmaster::PostmasterDefaultPriority - defines the default priority for all incoming emails.
- Ticket:Frontend::Agent:Ticket::ViewPhoneNew:Priority - defines the default priority in the New Phone Ticket screen for the agents.
- Ticket:Frontend::Agent:Ticket::ViewEmailNew:Priority - defines the default priority in the New Email Ticket screen for the agents.
- Ticket:Frontend::Customer:Ticket::ViewNew:PriorityDefault - defines the default priority in the New Ticket screen in the Customer frontend.

Chapter 14. Creating your own themes

You can create your own themes for OTRS to use the layout you like in the web frontend. To create own themes you should customize the output templates to your needs.

More information about the syntax and structure of output templates can be found in the Developer Manual on <http://doc.otrs.org> (<http://doc.otrs.org>) , especialy in the chapter about *templates* (<http://doc.otrs.org/developer/2.4/en/html/c629.html>) .

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

1. Create a directory called `Kernel/Output/HTML/Company` and copy all files, that you like to change, from `Kernel/Output/HTML/Standard` into the new folder.

Important: Only copy over the files you actually change. OTRS will automatically get the missing files from the Standard theme. This will make upgrading in a later stage much easier.

2. Customize the files in the directory `Kernel/Output/HTML/Company` and change the layout to your needs
3. To activate the new theme in OTRS, add them in the SysConfig under `Frontend::Themes` .

Now the new theme should be useable in OTRS. You can select it via your `personal preferences` page.

Warning

Do not change the theme files shipped with OTRS, since these changes will be lost after an update. Create your own themes only by performing the steps described above.

Chapter 15. Localization of the OTRS frontend

The frontend of OTRS supports different languages.

How localization for the OTRS framework is done, and what steps you have to perform to create a translation for a new language, or how you can create customizations to translations can be found in the "Language Translations" (<http://doc.otrs.org/developer/2.4/en/html/c772.html>) chapter from the developer manual on <http://doc.otrs.org> (<http://doc.otrs.org>) .

Chapter 16. PGP

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

1. Install GnuPG, via the package manager of your operating system.
2. Configure GnuPG for the usage with OTRS. The needed directories for GnuPG and a private key have to be created. The command shown in the Sript 16.1 has to be executed as OTRS user, from a shell.

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

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

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

```

"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Ticket System
Email address: support@example.com
Comment: Private PGP Key for the ticket system with address support@example.com
You selected this USER-ID:
"Ticket System (Private PGP Key for the ticket system with address support@exampl
le.com) <support@example.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.

Passphrase: secret
Repeat passphrase: secret

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
+++++++.+++++++.+++++++.+++++++.+++++++.+++++++.+++++++.+++++++.+++++++.+++++++.
+++++++.+++++++.+++++++.+++++++.+++++++.+++++++.+++++++.+++++++.+++++++.+++++++.
.....>+++++<+++++.....+++++

Not enough random bytes available. Please do some other work to give
the OS a chance to collect more entropy! (Need 280 more bytes)

+++++++.+++++..+++++++.+++++...+++++++.+++++++.+++++++.+++++++.+++++++.+++++
+++++++.+++++++.+++++++.+++++++.+++++++.+++++++.+++++++.+++++++.+++++++.+++++
..+++++>..+++++...>+++++.....>+++++<+++++.....
.....+++++^^^
gpg: /opt/otrs/.gnupg/trustdb.gpg: trustdb created
gpg: key 7245A970 marked as ultimately trusted
public and secret key created and signed.

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

linux:~$

```

Script 16.1. Configuring GnuPG.

As shown in the Script 16.1, for most of the settings the default is OK. Only the values for the key owner have to be entered correctly and a proper password for the key has to be specified.

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

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

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

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

How to get the id of your own private key? The ID of your own private key is already shown during the key generation (see step 1 from above). It is also possible to get the ID if the command specified in the Script 16.2 is executed as OTRS user:

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

linux:~$
```

Script 16.2. Getting the ID of your own private key.

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

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

4. Finally, import a customer's public key. This ensures that also encrypted messages can be sent out to this customer. There are two ways to import a public key of a customer.

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

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

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

Chapter 17. S/MIME

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

The S/MIME configuration is conducted outside the OTRS web interface for the most part, and should be carried out in a shell by the OTRS user, i.e. with his/her user rights. The MIME configuration under Linux is based on SSL (OpenSSL). Therefore, check first of all whether the OpenSSL package is installed on your system. The OpenSSL package includes a script called CA.pl, with which the most important steps of certificate creation can be performed. To simplify the procedure, find out where in the file system the CA.pl script is stored and enter the location temporarily into the path variable of the shell (see Script 17.1 below).

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

Script 17.1. Configuring S/MIME.

The Script 17.1 shows that a new temporary directory ~/tmp has been created, in which the certificate is generated.

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

1. Establish your own Certification Authority for SSL. You need it to certify the request for your own SSL certificate (see Script 17.2 below).

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

Making CA certificate ...
Generating a 1024 bit RSA private key
```

```

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

```

Script 17.2. Establishing a Certification Authority for SSL.

2. Generate a certificate request (see Script 17.3 below).

```

otrs@linux:~/tmp> CA.pl -newreq
Generating a 1024 bit RSA private key
.....+++++
....+++++
writing new private key to 'newreq.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.

```

```

-----
Country Name (2 letter code) [AU]:DE\keyreturn
State or Province Name (full name) [Some-State]:OTRS-state
Locality Name (eg, city) []:OTRS-town
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Your company
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:OTRS admin
Email Address []:otrs@your-domain.tld

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

```

Script 17.3. Creating a certificate request.

3. Signing of the certificate request. The certificate request can either be signed, and thereby certified, by your own CA or, which is more respectable, by another extern certified CA (see Script 17.4 below).

```

otrs@linux:~/tmp> CA.pl -signreq
Using configuration from /etc/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number:
        fd:85:f6:9f:14:07:16:c8
    Validity
        Not Before: Jan  8 17:04:37 2006 GMT
        Not After : Jan  8 17:04:37 2007 GMT
    Subject:
        countryName           = DE
        stateOrProvinceName   = OTRS-state
        localityName          = OTRS-town
        organizationName      = Your Company
        commonName            = OTRS administrator
        emailAddress          = otrs@your-domain.tld
X509v3 extensions:
    X509v3 Basic Constraints:
        CA:FALSE
    Netscape Comment:
        OpenSSL Generated Certificate

```

```

X509v3 Subject Key Identifier:
    01:D9:1E:58:C0:6D:BF:27:ED:37:34:14:D6:04:AC:C4:64:98:7A:22
X509v3 Authority Key Identifier:
    keyid:10:4D:8D:4C:93:FD:2C:AA:9A:B3:26:80:6B:F5:D5:31:E2:8E:DB:A8
    DirName:/C=DE/ST=OTRS-state/L=OTRS-town/O=Your Company/
    CN=OTRS admin/emailAddress=otrs@your-domain.tld
    serial:FD:85:F6:9F:14:07:16:C7

```

```

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

```

```

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

```

Script 17.4. Signing of the certificate request.

4. Generate your own certificate, and all data going with it, using the signed certificate request (see Script 17.5 below).

```

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

```

Script 17.5. Generating a new certificate.

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

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

With the SysConfig group "Crypt::SMIME" , you can also enable and configure the general S/MIME support.

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

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

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

Chapter 18. Access Control Lists (ACLs)

From OTRS 2.0 on, Access Control Lists (ACLs) can be used to control access to tickets, modules, queues, etc. or to influence actions on tickets (closing, moving, etc.) in certain situations. ACLs can be used supplementary to the existing permission system of roles and groups. Using ACLs, rudimental workflows within the system can be mapped, based on ticket attributes.

So far, ACLs cannot be created using the SysConfig interface but must be directly entered into the `Kernel/Config.pm` file. Find some examples below (Example 18-1, 18-2 and 18-3):

Example 18-1. ACL which only allows to move tickets with ticket priority 5 into a queue.

```
# ticket acl
$Self->{TicketAcl}->{'ACL-Name-2'} = {
    # match properties
    Properties => {
        # current ticket match properties
        Ticket => {
            Queue => ['Raw'],
            Priority => ['5 very high'],
        }
    },
    # return possible options (white list)
    Possible => {
        # possible ticket options (white list)
        Ticket => {
            Queue => ['Alert'],
        },
    },
};
```

Example 18-2. ACL which disables the closing of tickets in the raw queue and hides the close button.

```
$Self->{TicketAcl}->{'ACL-Name-1'} = {
    # match properties
    Properties => {
        # current ticket match properties
        Ticket => {
            Queue => ['Raw'],
        }
    },
    # return possible options (white list)
    Possible => {
        # possible ticket options (white list)
```

```

Ticket => {
    State => ['new', 'open', 'pending reminder'],
},
# possible action options
Action => {
    AgentTicketLock => 1,
    AgentTicketZoom => 1,
    AgentTicketClose => 0,
    AgentTicketPending => 1,
    AgentTicketNote => 1,
    AgentTicketHistory => 1,
    AgentTicketPriority => 1,
    AgentTicketFreeText => 1,
    AgentTicketHistory => 1,
    AgentTicketCompose => 1,
    AgentTicketBounce => 1,
    AgentTicketTicketPrint => 1,
    AgentTicketForward => 1,
    AgentTicketTicketLink => 1,
    AgentTicketPrint => 1,
    AgentTicketPhone => 1,
    AgentTicketCustomer => 1,
    AgentTicketOwner => 1,
},
},
};

```

Example 18-3. ACL which removes the status for all agents and only provides it for a group.

```

$Self->{TicketAcl}->{'ACL-Name-5'} = {
    # match properties
    Properties => {
        # current ticket match properties (match always)
    },
    # return possible options
    PossibleNot => {
        # possible ticket options
        Ticket => {
            State => ['closed successful'],
        },
    },
};

```

In the Script 18.1 there is a list of all parameters which can be used for ACLs.

```

# ticket acl
$Self->{TicketAcl}->{'ACL-Name-Test'} = {
    # match properties
    Properties => {
        # current action match properties
        Frontend => {
            Action => ['AgentTicketPhone', 'AgentTicketEmail'],
        },
        # current user match properties
        User => {
            Group_rw => [
                'hotline',
            ],
        },
        # current user match properties
        Ticket => {
            Queue => ['Raw'],
            State => ['new', 'open'],
            Priority => ['some priority'],
            Lock => ['lock'],
            CustomerID => ['some id'],
            CustomerUserID => ['some id'],
            TicketFreeKey1 => ['some key'],
            TicketFreeKey2 => ['some key'],
            # ...
            TicketFreeKey8 => ['some key'],
            TicketFreeText1 => ['some value'],
            TicketFreeText2 => ['some value'],
            # ...
            TicketFreeText8 => ['some value'],
        }
    },
    # return possible options (white list)
    Possible => {
        # possible ticket options (white list)
        Ticket => {
            Queue => ['Hotline', 'Koordination'],
            State => => ['some state'],
            Priority => ['5 very high'],
            TicketFreeKey1 => ['some key'],
            TicketFreeKey2 => ['some key'],
            # ...
            TicketFreeKey8 => ['some key'],
            TicketFreeText1 => ['some value'],
            TicketFreeText2 => ['some value'],
            # ...
            TicketFreeText8 => ['some value'],
        },
        # possible action options (white list)
        Action => {
            AgentTicketLock => 1,
            AgentTicketZoom => 1,
            AgentTicketClose => 1,

```

```

        AgentTicketPending => 0,
        AgentTicketNote => 1,
        AgentTicketHistory => 0,
        AgentTicketPriority => 1,
        AgentTicketFreeText => 0,
        AgentTicketHistory => 1,
        AgentTicketCompose => 1,
        AgentTicketBounce => 1,
        AgentTicketTicketPrint => 0,
        AgentTicketForward => 1,
        AgentTicketTicketLink => 1,
        AgentTicketPrint => 1,
        AgentTicketPhone => 1,
        AgentTicketCustomer => 1,
        AgentTicketOwner => 0,
    },
},
# remove options (black list)
PossibleNot => {
    # possible ticket options (black list)
    Ticket => {
        Queue => ['Hotline', 'Koordination'],
        State => ['closed', 'removed'],
    },
},
};

```

Script 18.1. Valid parameters for ACLs.

Chapter 19. Stats module

Stats and reports are essential for companies. On the basis of stats, information can be evaluated and rated, and eventually informed decisions can be made.

A stats module should meet a multitude of requirements:

- Various OTRS modules are to be evaluated.
- Consideration of the user permission settings of the respective installation.
- Generation of complex statistics.
- Easy and fast configuration of stats.
- High performance.
- Support of a large number of output variants.

The OTRS stats module tries to satisfy all these requirements. The core feature is a GUI, which allows to configure stats with the click of a mouse.

Statistical elements, i.e. files which supplement the functionalities of the stats module for specific requirements, can be integrated to model complex stats.

All requirements concerning the evaluation of personal data have been considered. It is, for instance, not possible to directly evaluate the activities of individual agents in the standard configuration.

The stats module enables OTRS users to make a quick and easy evaluation of the activities on their system.

19.1. Handling of the module by the agent

As soon as the stats module has been installed and configured successfully by the OTRS admin, the navigation bar of every agent with read and/or write rights displays the additional module link "Statistics", which has various action links, as shown in the Figure 19.1.

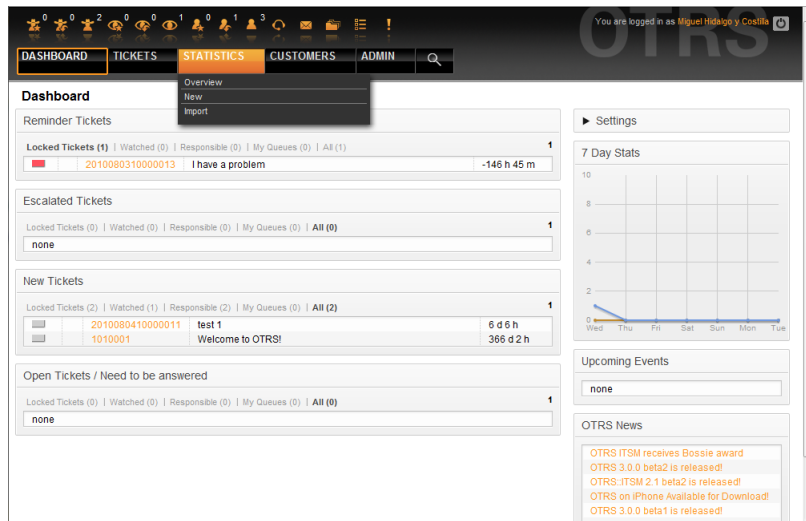


Figure 19.1. Action lists of the statistics.

The different options provided in the statistics menu are:

- *Overview*. Automatically shown if the module link "Statistics" is chosen.
- *New*. Requires rw rights.
- *Import*. Requires rw rights.

19.1.1. Overview

A list of all pre-configured stats the agent can use (see Figure 19.2 below).

The easiest way to access the overview is to select the "Statistics" link and then the action link "Overview" in the navigation bar. The stats overview is called up automatically when the module link "Statistics" in the navigation bar is chosen.

STATE	TITLE	OBJECT	DESCRIPTION
10001	List of the most time-consuming tickets	Ticketlist	List of tickets closed last month which required ti...
10002	Changes of status in a monthly overview		Monthly overview, which reports status changes per[...]
10003	List of tickets created last month	Ticketlist	List of all tickets created last month. Order by al...
10004	List of open tickets, sorted by time left until solution deadline expires	Ticketlist	List of open tickets, sorted by time left until so[...]
10005	List of tickets closed, sorted by solution time	Ticketlist	List of tickets closed last month, sorted by solu[...]
10006	List of open tickets, sorted by time left until escalation deadline expires	Ticketlist	List of open tickets, sorted by time left until es[...]
10007	List of tickets closed last month	Ticketlist	List of all tickets closed last month. Order by ag[...]
10008	Overview about all tickets in the system	TicketAccumulation	Current state of all tickets in the system without[...]
10009	List of tickets closed, sorted by response time	Ticketlist	List of tickets closed last month, sorted by respo[...]
10010	List of open tickets, sorted by time left until response deadline expires	Ticketlist	List of open tickets, sorted by time left until rei...
10011	New Tickets	TicketAccumulation	Total number of new tickets per day and queue whic[...]

Figure 19.2. Overview of the standard statistics.

The following information is provided about the stats listed in the overview:

- *Stat#*. Unique statistic number.
- *Title*. Title of the statistic.
- *Object*. Object used for generating the statistic. In the case of a static statistic, no object is displayed as no dynamic object is used for its generation.
- *Description*. Excerpt of the statistic's description.

Below the list of statistics, there is a results display, which provides information about the number of accessible stats. If the overview covers more than one page, the agent can browse through the different pages.

The agents can sort the stats ascending or descending, according to their needs, just by clicking the desired sorting key field of the headline.

To generate a statistic, the right one is chosen by clicking the link with the stat number, which leads to the "View" user interface.

A few sample stats are imported when the stats module is used for the first time.

19.1.2. Generate and view stats

The view user interface is supposed to provide the agent with an overview of the stat's pre-configuration (see Figure 19.3 below).

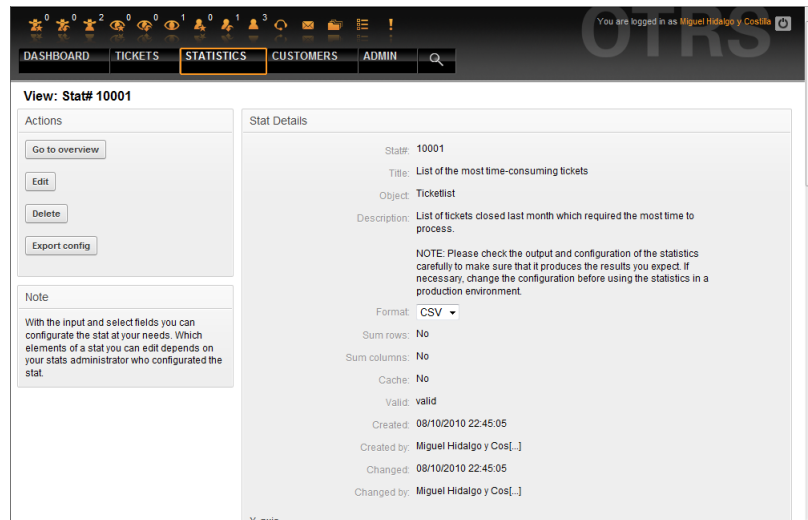


Figure 19.3. Viewing a specific statistic.

The agents can perform configurations within the range of the pre-configuration. The originator of a stat configuration can determine the degree to which the stat generation can be influenced. The most extreme examples of this would be agents who can't effect any changes or can change all attributes, respectively.

The page shows the following:

- Possible actions:
 - *Go to overview*. Link back to the starting point.
 - *Edit*. Edit the current stat (rw rights required).
 - *Delete*. Delete the current stat (rw rights required).
 - *Export config*. Export a stat configuration, via file download (rw rights required).

Usage: export and import functions allow for the convenient creation and testing of stats on test systems and subsequent easy integration into the production system.

- The stat details:
 - *Stat#*. The number of the statistic.
 - *Title*. The title of the statistic.
 - *Object*. Object used for generating the stat.
 - *Description*. Provides more detailed information about the intention of the stat.
 - *Format*. Stat output format. Depending on the configuration, any of the following output formats can be chosen:
 - CSV.
 - Print.
 - Graph-lines.
 - Graph-bars.
 - Graph-hbars.
 - Graph-points.
 - Graph-lines-points.
 - Graph-area.
 - Graph-pie.
 - *Graphsize*. Size (in pixels) which the graphic / chart may have. This option is only given when the stat pre-configuration allows it for a chart. All generally usable graphic sizes are configured by the OTRS admin in the SysConfig. The agent can then pre-select all relevant formats, while configuring the stats.
 - *Sum rows*. Indicates whether the stat is amended by a column, whose cells state the sums of the respective rows.
 - *Sum columns*. Indicates whether the stat is amended by a row, whose cells state the sum of the respective columns.
 - *Cache*. Indicates whether the generated stats are cached in the file system.
 - *Valid*. This is set to "invalid" if a pre-configured statistic is still being created or must not be used temporarily for certain reasons. Also the "Start" button in the bottom right of the block is then no longer displayed. The stat can no longer be generated.
 - *Created*. Creation time of the stat.
 - *Created by*. Agent who created the stat.

- *Changed*. Time of the last stat's modification.
- *Changed by*. Agent who altered the stat.
- *X-axis*. Using this function, the agent can exchange the x and y axes (only when activated by the OTRS administrator).
- The general information is followed by information about the stat itself. There are two different stat views:
 - *Static stat view*. Complex stats cannot be generated dynamically, that is the reason why static stat generators can be integrated into the stat module (see Figure 19.4 below). They were developed specifically for the particular evaluation. The various job definitions result in different views.

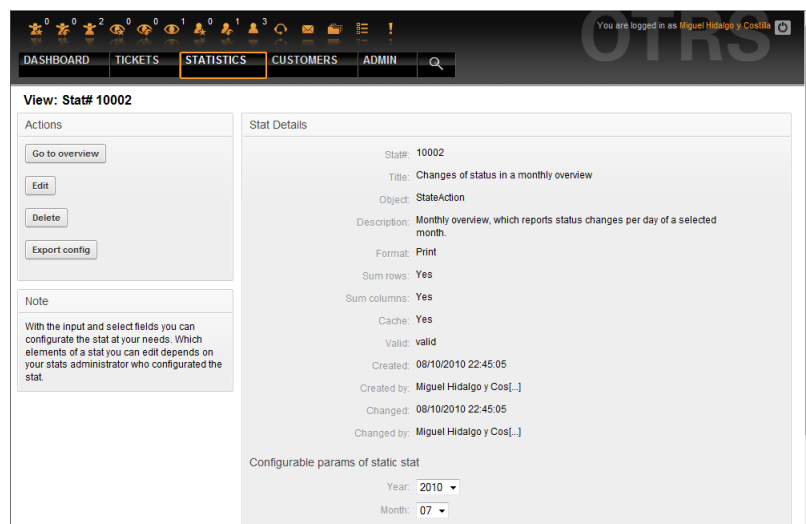


Figure 19.4. Viewing a static statistic.

- *Dinamic stat view* (see Figure 19.3 above). They can be displayed in two different ways:
 - *Unchangeable settings*. The originator of the stat has no permission for modifying this fields.
 - *Changeable settings*. The pre-configuration of such stats can be changed by the agent.

Pressing the "Start" button (at the bottom of the screen) is the last simple step to generate a stat. There are two possible reasons for this button to not be displayed:

1. The stat was set to invalid and thus, deactivated.
2. The stat was not configured neatly and is, therefore, not yet executable. In this case, the necessary information can be found in the OTRS notification section (below the navigation bar).

If the settings on the view page are incorrect, this page is shown again after the "Start" button was pushed, and information about which input was incorrect is provided in the notification section.

19.1.3. Edit - New

Agents, who have write rights, can edit an existing stat configuration by calling up the edit user interface of the stat module. There are two possibilities for that:

1. Via the "Edit" button in the stat view.
2. Via the "New" link in the menu of the Statistics module in the navigation bar.

The stats are edited with a wizard in four steps:

1. General specifications.
2. Definition of the element for the X-axis.
3. Specification of the value series.
4. Definition of the restrictions to characterize the stat.

Points 2 to 4 are only needed for the generation of dynamic stats. For a static stat, only the general information (point 1) is required.

Information about how to handle the page is provided below the input forms, on all input user interfaces of the stats module.

If incorrect data is entered nevertheless, the previously processed user interface is displayed again and amended by information about the incorrect input. This information can be found in the OTRS notification section. The next input user interface is only displayed after the form has been filled out correctly.

1. *General specifications.* It is the first page of the wizard (see Figure 19.5 below).

Figure 19.5. Editing the general specifications of a stat.

In the screen showed in the Figure 19.5 there is a great number of common specifications and settings can be edited:

- *Title.* Should reflect the stat's purpose in a concise manner.
- *Description.* More detailed information about the job definition, type of configuration parameters, etc.
- *Dynamic object.* If the OTRS installation provides various dynamic objects, one of them can be chosen. The objects meet the requirements of the particular modules.
- *Static file.* Usually this selection is not shown, as only static files which are not yet assigned to any stats are displayed. If "Static file" is displayed, however, it is important to tick the option field and select a generation mode (dynamic with a dynamic object or static with a file). If a static file is selected, the input user interfaces 2-4 are not shown as the static file contains all required configurations.

- *Permission settings.* Facilitate a restriction of the groups (and therefore, agents) who can later view and generate the preconfigured stats. Thus the various stats can be allocated to the different departments and work groups who need them. It is possible to allocate one stat to various groups.

Example 1: The "stats" group was selected. The stat is viewable for all users having at least ro rights for the "stats" group. As the base requirement for accessing the view mode of the stat module in the default configuration is to have ro rights in the "stats" group, all users with permission to create statistics can view it in this configuration.

Example 2: A group named "sales" was selected. All users with ro rights for the "sales" group can see the stat in the view mode and generate it. However, the stat will not be shown to other users with permission to generate stats.

- *Format.* Output format of the stat. Depending on the configuration, one or more of the following formats can be chosen:
 - CSV.
 - Print.
 - graph-lines.
 - graph-bars.
 - graph-hbars.
 - graph-points.
 - graph-lines-points.
 - graph-area.
 - graph-pie.
- *Graphsize.* Select the size (in pixels) the charts should have. This selection is only necessary if a graphical output format has been chosen under "Format". All graphic sizes that can generally be used are defined by the OTRS admin in SysConfig. When configuring the stat, the agent can pre-select all relevant formats.
- *Sum rows.* Indicates whether the stat is amended by a column, whose cells contain the sum of the respective row.
- *Sum columns.* Indicates whether the stat is amended by a row, whose cells contain the sum of the respective column.
- *Cache.* Specifies if the generated stats should be cached in the file system. This saves computing power and time if the stat is called up again, but it only should be used if the stat's content is no longer changing.

Caching is automatically prevented if the stat contains no time designation values (new data is added continuously) or if a time designation value points to the future.

If a cached stat is edited, all cached data is deleted.

- *Valid*. It is set "invalid" if a pre-configured stat is still being created or must not be used temporarily for certain reasons. Also the "Start" button in the bottom right of the block is no longer displayed. It is impossible to generate the stat.

2. *Definition of the element for the X-axis*. It is the configuration of the element used for the depiction of the X-axis or, if tables are used, of the column name chosen for the stat (see Figure 19.6).

OTRS

You are logged in as Miguel Hidalgo y Cosío

DASHBOARD TICKETS **STATISTICS** CUSTOMERS ADMIN

Edit: Stat# 10001

1 2 3 4

Actions

Go to overview

Note

Here you can define the x-axis. You can select one element via the radio button. If you make no selection all attributes of the element will be used if you generate a stat, as well as new attributes which were added since the last configuration.

If you remove the hook in the "Fixed" checkbox, the agent generating the stat can change the attributes of the corresponding element.

Select the element that will be used at the X-axis (2/4)

Attributes to be printed

Number
Ticket#
Age
Title
Created
☒ Fixed

Back Next...

Powered by OTRS 3.0.x CVS

Top of page

Figure 19.6. Definition of the element for the X-axis.

First of all, an element is selected using the option field. Then two or more attributes of the element must be selected. If no attributes are selected, all attributes are used including those added after the configuration of the stat.

If the "Fixed" setting is disabled, the agent generating the stat can change the attributes of the respective element in the "View" user interface.

Time elements are different as time period and scale have to be stated. Type and number of elements result from the used dynamic object and vary depending on it.

If all input is correct, the "Next" button leads to the "Value series" form. It is also possible to go back to editing the "common statements" section again.

3. Specification of the value series.

In the third step of the stat pre-configuration the value series are defined (see Figure 19.7 below). They will later on form the individual graphs of a diagram or the various series (tabular view).

Edit: Stat# 10001

1 — 2 — 3 — 4

Actions

[Go to overview](#)

Note

Here you can define the value series. You have the possibility to select one or two elements. Then you can select the attributes of elements. Each attribute will be shown as single value series. If you don't select any attribute all attributes of the element will be used if you generate a stat, as well as new attributes which were added since the last configuration. If you remove the hook in the "Fixed" checkbox, the agent generating the stat can change the attributes of the corresponding element.

Select the elements for the value series (3/4)

☒ Order by: State
Priority
Customer User
CustomerID
Accounted time

☒ Sort sequence: ascending
Descending

☒ Fixed

[Back](#) [Next...](#)

Powered by OTRS 3.0.x CVS Top of page

Figure 19.7. Definition of the value series.

If an element is selected, each chosen attribute will correspond to a value series (see the Example 19-1 below).

Example 19-1. Definition of a value series - one element

Element Queue:

- Value series 1 = Raw
- Value series 2 = Junk
-

If two elements are selected, each selected attribute of the first element is combined with an attribute of the second element to form a value series (see the Example 19-2 below).

Example 19-2. Definition of a value series - two elements

Element 1 queue and element 2 status:

- Value chain 1 = Raw - open
- Value series 2 = Raw - successfully closed
- Value series 3 = Junk - open
- Value series 4 = Junk - successfully closed

Selection of three or more elements does not make sense and is, therefore, prevented by an error message.

Additionally the same conditions apply to the selection of the attributes and the "Fixed" checkbox as to the "X-axis" selection:

- If no attributes of the element are selected, all attributes are used, including those added after the configuration of the stat.
- If the "Fixed" setting is disabled, the agent generating the stat can change the attributes of the respective element.

4. *Definition of the restrictions to characterize the stat.* It is the fourth and last step of the pre-configuration (see Figure 19.8 below). The restrictions facilitate to limit the results and are comparable to entering data into a search window. In quite a few cases, no restrictions at all must be set up.

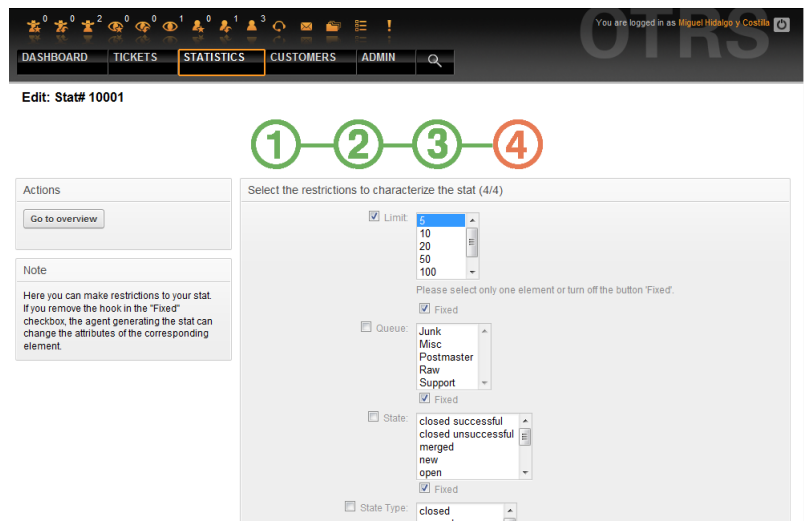


Figure 19.8. Definition of restrictions.

After all restrictions are set, the pre-configuration of the stat is finished with the "Finish" button and the "View" user interface is called up.

19.1.4. Import

The import user interface (see Figure 19.9 below) can be accessed by choosing "Import" in the navigation bar (rw rights required).

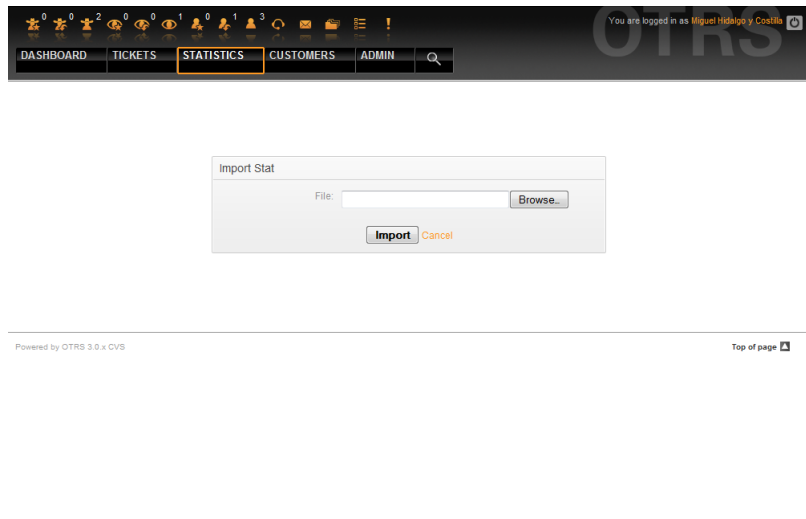


Figure 19.9. The import user interface.

Facilitates the import of stats and is, combined with the export function of the module, a very handy functionality. Stats can be created and tested conveniently on test systems and then imported into the production system.

The import is effected by an easy file upload. The "view" user interface of the imported stat is opened automatically afterwards.

19.2. Administration of the stat module by the OTRS administrator

This section provides information about the tasks and possibilities of the OTRS administrator dealing with the statistics module.

19.2.1. Permission settings, groups and queues

No new queues and/or groups are created when the stat module is installed.

The default configuration of the module registration gives all agents with "stats" group permissions access to the stats module.

Access according to permission settings:

- *rw*. Allows configuring statistics.
- *ro*. Permits generating pre-configured statistics.

The OTRS administrator decides whether agents with the permission to generate pre-configured stats are allocated *ro* rights in the "stats" group, or if their respective groups are added in the module registration in SysConfig.

19.2.2. SysConfig

SysConfig groups `Framework:Core::Stats` , `Framework:Core::Stats::Graph` and `Framework:Frontend::Agent::Stats` contain all configuration parameters for the basic set-up of the statistics module. Moreover, the configuration parameter `$Self->{'Frontend::Module'}->{'AgentStats'}` controls the arrangement and registration of the modules and icons within the statistics module.

19.3. Administration of the stats module by the system administrator

Generally, no system administrator is needed for the operation, configuration and maintenance of the statistics module. However, a little background information for the system administrator is given at this point.

Note: File paths refer to subdirectories of the OTRS home directory (in most cases `/opt/otrs`).

19.3.1. Data base table

All stat configurations are implemented and administrated in XML and, therefore, stored in the database table "xml_storage". Other modules whose content is presented in xml format use this table as well.

19.3.2. List of all files

The following files are necessary for the stats module to work accurately:

- Kernel/System/Stats.pm
- Kernel/Modules/AgentStats.pm
- Kernel/System/CSV.pm
- Kernel/Output/HTML/Standard/AgentStatsOverview.dtl
- Kernel/Output/HTML/Standard/AgentStatsDelete.dtl
- Kernel/Output/HTML/Standard/AgentStatsEditSpecification.dtl
- Kernel/Output/HTML/Standard/AgentStatsEditRestrictions.dtl
- Kernel/Output/HTML/Standard/AgentStatsEditXaxis.dtl
- Kernel/Output/HTML/Standard/AgentStatsEditValueSeries.dtl
- Kernel/Output/HTML/Standard/AgentStatsImport.dtl
- Kernel/Output/HTML/Standard/AgentStatsPrint.dtl
- Kernel/Output/HTML/Standard/AgentStatsView.dtl
- Kernel/System/Stats/Dynamic/Ticket.pm
- bin/otrs.GenerateStats.pl

19.3.3. Caching

Whether the results of a statistic are to be cached or not can be defined in the pre-configuration. Cached statistic results are stored as files in the `var/tmp` directory of the OTRS installation (in most cases `opt/otrs/var/tmp`).

Cached stats can be recognized by the "Stats" prefix.

If the data is lost, no major damage is caused. The next time the stat is called up, the stats module will not find the file any more and thus, will generate a new stat, which will probably take a little longer.

19.3.4. otrs.GenerateStats.pl

This file is saved in the `bin/` directory. It facilitates the generation of stats in the command line.

The command line call in the Script 19.1 is an exemplary.

```
bin> perl otrs.GenerateStats.pl -n 10004 -o /output/dir
```

Script 19.1. Generating a stat via command line.

A stat from the stat configuration "Stat# 10004" is generated and saved as csv in the `/output/dir` directory.

The generated stat can also be sent as an e-mail. More information can be called up with the command of the Script 19.2.

```
bin> perl otrs.GenerateStats.pl --help
```

Script 19.2. Getting information about the `otrs.GenerateStats.pl` file.

19.3.5. Automated stat generation - Cronjob

It obviously does not really make sense to generate stats manually in the command line, as the stat module has a convenient graphical user interface. However, generating stats manually does make sense when combined with a Cronjob.

Imagine the following scenario: On every first day of the month the heads of department want to receive a stat for the past month. By combining a cronjob and command line call the stats can be sent to them automatically by e-mail.

19.3.6. Static stats

The stats module facilitates the generation of static statistics. For every static stat a file exists in which its content is precisely defined.

This way, very complex stats can be generated. The disadvantage is that they are not particularly flexible.

The files are saved in the following directory `Kernel/System/Stats/Static/`.

19.3.7. Using old static stats

Standard OTRS versions 1.3 and 2.0 already facilitated the generation of stats. Various stats for OTRS versions 1.3 and 2.0 which have been specially developed to meet customers' requirements can be used in more recent versions too.

The files must merely be moved from the `Kernel/System/Stats/` path to `Kernel/System/Stats/Static/`. Additionally the package name of the respective script must be amended by `::Static`.

The following example shows how the first path is amended.

```
package Kernel::System::Stats::AccountedTime;
```

```
package Kernel::System::Stats::Static::AccountedTime;
```

19.3.8. Default stats

"It is not always necessary to reinvent the wheel..."

The stats module provides various default stats. Stats which are interesting for all OTRS users will in future be added to the default stats of the stats module package. Default stats are saved in the stats module xml format in the `scripts/test/sample/` directory.

Chapter 20. Additional applications

You can install additional application packages to extend the functionality of the OTRS framework. This can be done via the package manager in the adminarea, which downloads the applications from an online repository and manages package dependencies. It is also possible to install packages from local files, in case your OTRS system does not have a direct internet connection.

20.1. Calendar

You can edit and view both private and public user-based appointments with the web-based Calendar (see Figure 20.1 below).

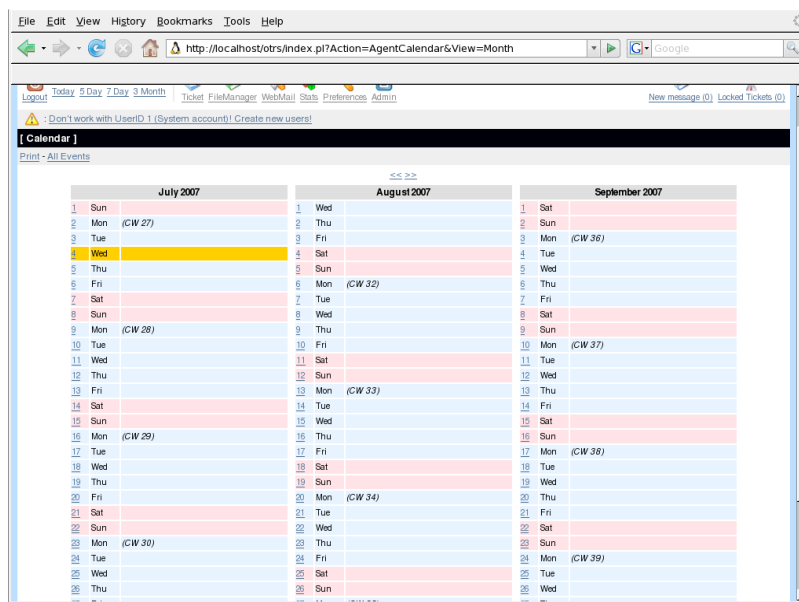


Figure 20.1. Calendar package.

20.2. File Manager

The File Manager (see Figure 20.2 below) makes it possible to access the content of a directory that is

located in the file system of the machine where OTRS is installed. You can use your web browser to upload, download, delete and view files, browse and create subdirectories, etc.

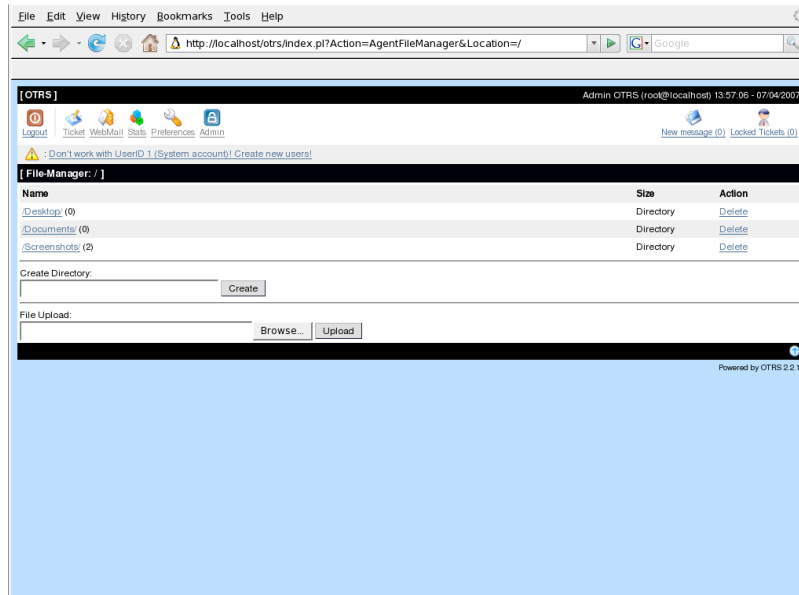


Figure 20.2. File Manager package.

20.3. Web Mail

Using the Web Mail package (see Figure 20.3 below), you can access your mailbox via an IMAP server. With a web browser, it is possible to compose new messages, answer and forward your mails, etc.

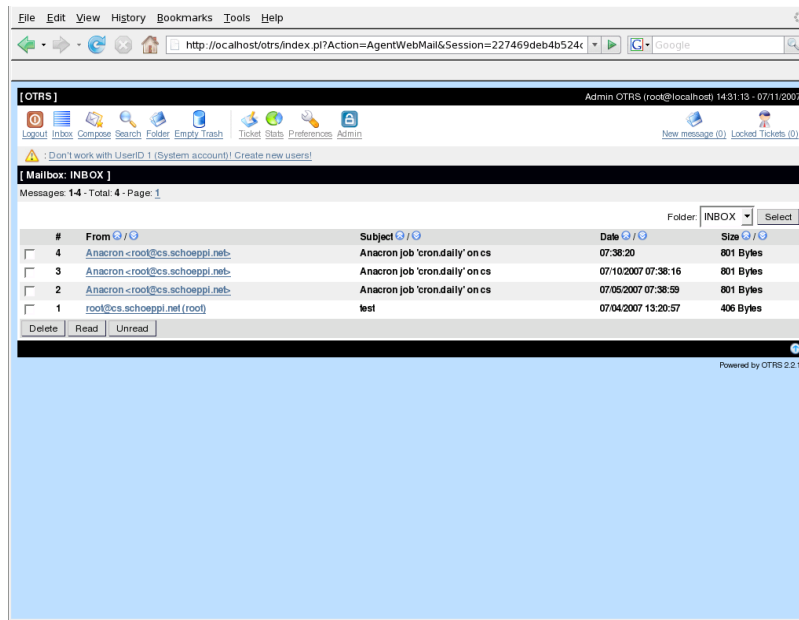


Figure 20.3. Web Mail module.

20.4. FAQ

It is the Knowledge Base component. It supports editing and viewing of FAQ articles. Articles can be viewed and restricted to agents, customer users or anonymous users; they can also be structured in groups and have different languages.

Chapter 21. 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.

21.1. OTRS

There are several options to improve the performance of OTRS.

21.1.1. TicketIndexModule

There are two backend modules for the ticket index:

- `Kernel::System::Ticket::IndexAccelerator::RuntimeDB` (default), generate each queue view on the fly from the ticket table. You will not have performance trouble until you have about 60,000 tickets (with 6,000 open tickets) in your system.
- `Kernel::System::Ticket::IndexAccelerator::StaticDB`, the most powerful module, it should be used when you have above 80,000 tickets (more than 6,000 open tickets) - it use an extra `ticket_index` table, which works like a view. Use `bin/otrs.RebuildTicketIndex.pl` for generating an initial index build after switching backends.

You can change the `IndexAccelerator` via `SysConfig`.

21.1.2. TicketStorageModule

There are two different backend modules for the ticket/article storage:

- `Kernel::System::Ticket::ArticleStorageDB` (default), store attachments, etc. in the database. Note: Don't use it for large set ups.

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

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

- Kernel::System::Ticket::ArticleStorageFS, store attachments etc. on the local file system. Note: Recommended for large setups.

Pro: Fast!

Contra: Your web server user should be the otrs user (file system permissions!). Also, if you have multiple front end servers, you should make sure the file system is shared between the servers. So place it on an NFS share or preferably a SAN or similar solution.

Note: you can switch from one to the other back-end on the fly. You can switch the backend in the SysConfig, and then you can run the command line utility `otrs.ArticleStorageSwitch.pl` to put the articles from the database onto the filesystem or the other way around. You can use the `-s` and `-d` option to specify the source and destination back-ends. Please note that the whole process can take a lot of time, depending on the amount of articles you have and the available CPU power and/or network capacity.

```
shell> bin/otrs.ArticleStorageSwitch.pl -s ArticleStorageDB -d ArticleStorageFS
```

Script 21.1 Switching storage back-ends from database to filesystem.

21.1.3. Archiving Tickets

As OTRS can be used as an audit proof system, deleting closed tickets may not be a good idea. Therefore we implemented a feature that allows you to archive tickets.

This means that tickets that match certain criteria are marked as "archived", and these tickets are not touched if you do a regular ticket search or run a Generic Agent job. The system itself has not to deal with a huge amount of tickets any longer as only the "latest" tickets are taken into consideration when using OTRS. This can mean a huge performance gain on large systems.

To use the archive feature simply follow these steps:

1. Activate the archive system in SysConfig

In the Admin panel of OTRS go to SysConfig and select the group `Ticket`. In `Core::Ticket` you find the option `Ticket::ArchiveSystem` which is set to "no" per default. Change this setting to "yes" and save this change.

2. Define a GenericAgent job

In the Admin panel select GenericAgent and add a new job there.

a. Job Settings

Provide a name for the archiving job and select proper options to schedule this job.

b. Ticket Filter

The ticket filter is basically a ticket search that is looking for tickets that match the selected criteria. It might be a good idea to only archive tickets in a close state that have been closed a few months before.

c. Ticket Action

In the section Ticket Action you will find an action called "Archive selected tickets". Set this to "archive tickets".

d. Save the job

At the end of the page you will find an option to save the job.

e. Affected tickets

The system will display all tickets which will be archived when executing the Generic Agent job

3. Ticket Search

When you search for tickets, the system default is only searching tickets which are not archived. If you want to search through archived tickets also, simply add "archive search" while defining search criteria.

21.2. Database

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

21.2.1. MySQL

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

You should try this if mysqld needs a lot of your CPU time. Optimize tables ticket, ticket_history and article (see Script 21.2 below).

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

Script 21.2. Optimizing data base tables.

21.2.2. 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.revsys.com/writings/postgresql-performance.html>
(<http://www.revsys.com/writings/postgresql-performance.html>)
- <http://varlena.com/GeneralBits/Tidbits/perf.html> (<http://varlena.com/GeneralBits/Tidbits/perf.html>)
- http://varlena.com/GeneralBits/Tidbits/annotated_conf_e.html
(http://varlena.com/GeneralBits/Tidbits/annotated_conf_e.html)

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

21.3. 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 it needs more RAM.

21.3.1. Pre-establish database connections

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

21.3.2. Preloaded modules - startup.pl

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

21.3.3. Reload Perl modules when updated on disk

By default Apache::Reload is used in `scripts/apache2-httpd.include.conf`. Disable it and you will get 8% more speed. But remember to restart the web server if you install any modules via the OTRS Package Manager, any values in your SysConfig or in Kernel/Config.pm. Important: this would also mean you can't use the OTRS Package Manager via the web interface, you need to use the command line variant - `bin/otrs.PackageManager.pl`.

21.3.4. Choosing the Right Strategy

If you have a really large installation (over 1,000 new tickets per 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>)).

21.3.5. mod_gzip/mod_deflate

If your bandwidth is a little bit small, use mod_deflate for Apache2. If you have an html page with 45k, mod_gzip/mod_deflate compresses it. The page will be about 7k. The drawback is that this increases load on the server side.

Chapter 22. Backing up the system

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

22.1. Back up

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

To simplify back ups, the script `scripts/backup.pl` is included in every OTRS installation, which backs up all important data (see Script 22.1 below).

```
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#
```

Script 22.1. Getting help about the back up mechanism.

Execute command specified in the Script 22.2 to create a back up:

```
linux:/opt/otrs/scripts# ./backup.pl -d /backup/
Backup /backup//2010-09-07_14-28/Config.tar.gz ... done
Backup /backup//2010-09-07_14-28/Application.tar.gz ... done
Dump MySQL rdbs ... done
Compress SQL-file... done
linux:/opt/otrs/scripts#
```

Script 22.2. Creating a back up.

All data was stored in the directory `/backup/2010-09-07_14-28/` (see Script 22.3 below). Additionally the data was saved into a `.tar.gz` file.


```
linux:/opt/otrs/scripts# ls /backup/2010-09-07_14-28/
Application.tar.gz  Config.tar.gz  DatabaseBackup.sql.gz
linux:/opt/otrs/scripts#
```

Script 22.3. Checking the back up files.

22.2. Restore

To restore a back up, the saved data of 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` (see Script 22.4 below) to simplify the restore process is shipped with every OTRS installation. It supports MySQL and PostgreSQL.

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

Script 22.4. Getting help about the restore mechanism.

Data, that is stored in the directory `/backup/2010-09-07_14-28/`, can be restored with the command specified in the Script 22.5 and an OTRS installation in `/opt/otrs`.

```
linux:/opt/otrs/scripts# ./restore.pl -b /backup/2010-09-07_14-28 -d /opt/otrs/
Restore /backup/2010-09-07_14-28//Config.tar.gz ...
Restore /backup/2010-09-07_14-28//Application.tar.gz ...
create MySQL
decompresses SQL-file ...
cat SQL-file into MySQL database
compress SQL-file...
linux:/opt/otrs/scripts#
```

Script 22.5. Restoring the system.

Appendix A. Additional Resources

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

A.1. Homepage OTRS.org

You can find our project homepage at <http://www.otrs.org/> (<http://www.otrs.org/>) (see Figure A-1 below).

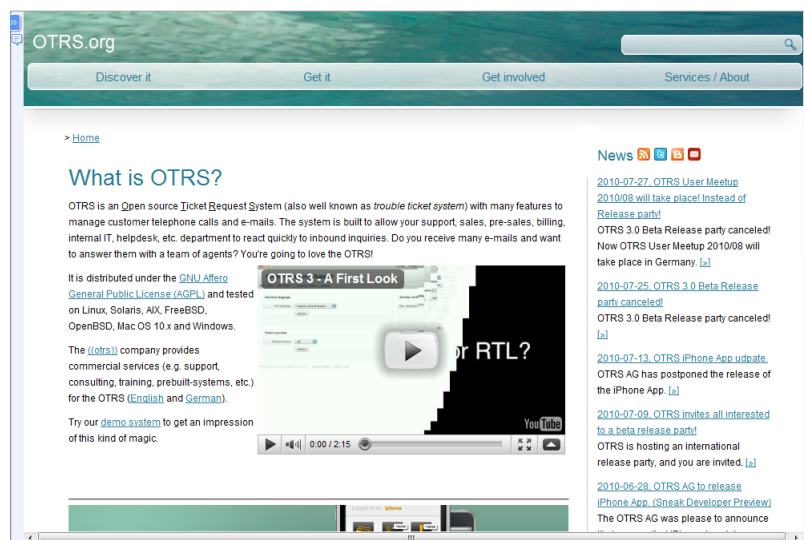


Figure A-1. OTRS Homepage.

A.2. Mailing lists

The Table A-1 shows you our different community mailing lists.

Table A-1. Mailinglists

Name	Description	Homepage
------	-------------	----------

Name	Description	Homepage
announce@otrs.org	Low traffic list, in English, for announcements of new OTRS releases and security issues.	http://lists.otrs.org/cgi-bin/listinfo/announce (http://lists.otrs.org/cgi-bin/listinfo/announce)
otrs@otrs.org	Medium to high traffic list, in English, where you can find all sorts of questions on, and support for the product.	http://lists.otrs.org/cgi-bin/listinfo/otrs (http://lists.otrs.org/cgi-bin/listinfo/otrs)
otrs-de@otrs.org	Medium to high traffic list, in German, where you can find all sorts of questions on, and support for, the product.	http://lists.otrs.org/cgi-bin/listinfo/otrs-de (http://lists.otrs.org/cgi-bin/listinfo/otrs-de)
dev@otrs.org	Medium traffic list, in English, where the OTRS developers discuss various design and implementation issues here.	http://lists.otrs.org/cgi-bin/listinfo/dev (http://lists.otrs.org/cgi-bin/listinfo/dev)
doc-de@otrs.org	Low traffic list, in German, with all sorts of questions on the documentation of the product.	http://lists.otrs.org/cgi-bin/listinfo/doc-de (http://lists.otrs.org/cgi-bin/listinfo/doc-de)
i18n@otrs.org	Low traffic list, in English, for internationalization and localization questions. If you are or want to become a translator of the OTRS project or have any problems with one of our applications in an international environment, this is the right place.	http://lists.otrs.org/cgi-bin/listinfo/i18n (http://lists.otrs.org/cgi-bin/listinfo/i18n)
cvs-log@otrs.org	Very high traffic list of CVS commit notifications.	http://lists.otrs.org/cgi-bin/listinfo/cvs-log (http://lists.otrs.org/cgi-bin/listinfo/cvs-log)

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

A.3. Bug tracking

To submit bugs visit <http://bugs.otrs.org/> (<http://bugs.otrs.org/>) (see Figure A-2 below). Please take note of the difference between a bug and a configuration issue. Configuration issues are problems that you encounter when setting up a system, or general questions on the use of OTRS. Bug reports should only be used for issues with the source code of OTRS itself, or to file enhancements for OTRS. All your bug reports and enhancement requests are very welcome in the bug tracker.

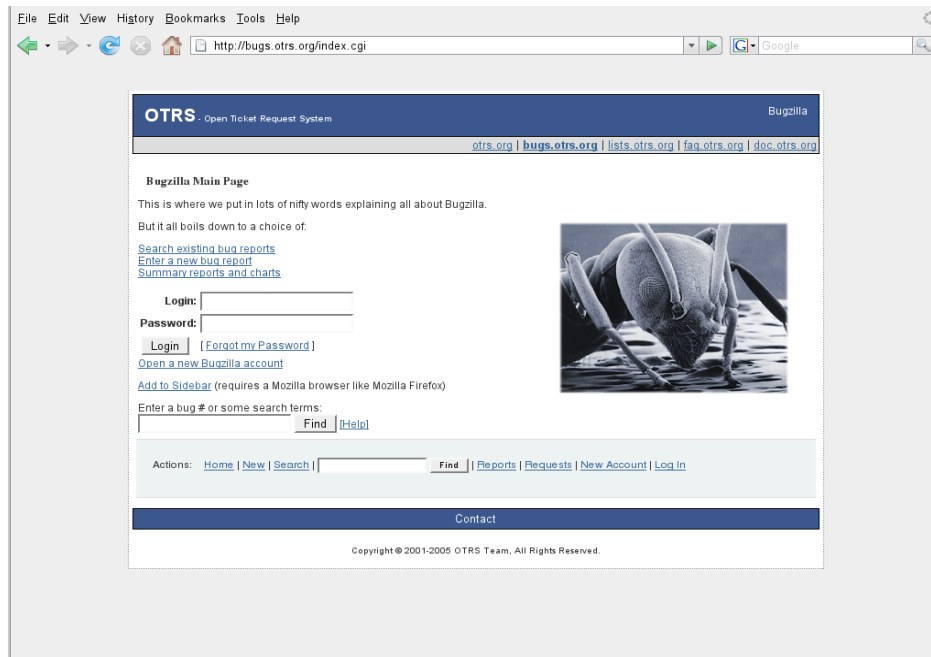


Figure A-2. Bug tracker.

For configuration issues you should either use the *commercial support*, available from *OTRS.com* (<http://www.otrs.com/en/support/>), or the public mailing lists.

You help us improving the product if you report bugs. We appreciate your input!

A.4. Commercial Support

For services (support, consulting, development, and training) you can contact the company behind OTRS, OTRS AG. They have offices in the Germany, US, Mexico and the Netherlands. Look at their

website for contact information: <http://www.otros.com/en/corporate-navigation/contact/>
(<http://www.otros.com/en/corporate-navigation/contact/>)

Appendix B. Configuration Options Reference

B.1. Framework

B.1.1. Core

B.1.1.1. SecureMode

Description	Value
Description:	Disables the web installer (http://yourhost.example.com/otrs/installer.pl), to prevent the system from being hijacked. If set to "No", the system can be reinstalled and the current basic configuration will be used to pre-populate the questions within the installer script. If not active, it also disables the GenericAgent, PackageManager and SQL Box (to avoid the use of destructive queries, such as DROP DATABASE, and also to steal user passwords).
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SecureMode'} = '0';</code>

B.1.1.2. Frontend::DebugMode

Description	Value
Description:	Enables or disable the debug mode over frontend interface.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Frontend::DebugMode'} = '0';</code>

B.1.1.3. ConfigLevel

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

Description	Value
Description:	Sets the configuration level of the administrator. Depending on the config level, some sysconfig options will be not shown. The config levels are in ascending order: Expert, Advanced, Beginner. The higher the config level is (e.g. Beginner is the highest), the less likely is it that the user can accidentally configure the system in a way that it is not usable any more.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'ConfigLevel'} = '100';</code>

B.1.1.4. ProductName

Description	Value
Description:	Defines the name of the application, shown in the web interface, tabs and title bar of the web browser.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'ProductName'} = 'OTRS';</code>

B.1.1.5. SystemID

Description	Value
Description:	Defines the system identifier. Every ticket number and http session string contain this ID. This ensures that only tickets which belong to your system will be processed as follow-ups (useful when communicating between two instances of OTRS).
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SystemID'} = '10';</code>

B.1.1.6. FQDN

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

Description	Value
Description:	Defines the fully qualified domain name of the system. This setting is used as a variable, OTRS_CONFIG_FQDN which is found in all forms of messaging used by the application, to build links to the tickets within your system.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'FQDN'} = 'yourhost.example.com';</code>

B.1.1.7. HttpType

Description	Value
Description:	Defines the type of protocol, used by the web server, to serve the application. If https protocol will be used instead of plain http, it must be specified it here. Since this has no affect on the web server's settings or behavior, it will not change the method of access to the application and, if it is wrong, it will not prevent you from logging into the application. This setting is used as a variable, OTRS_CONFIG_HttpType which is found in all forms of messaging used by the application, to build links to the tickets within your system.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'HttpType'} = 'http';</code>

B.1.1.8. ScriptAlias

Description	Value
Description:	Sets the prefix to the scripts folder on the server, as configured on the web server. This setting is used as a variable, OTRS_CONFIG_ScriptAlias which is found in all forms of messaging used by the application, to build links to the tickets within the system.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'ScriptAlias'} = 'otrs/';</code>

B.1.1.9. AdminEmail

Description	Value
Description:	Defines the system administrator's email address. It will be displayed in the error screens of the application.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'AdminEmail'} = 'admin@example.com';</code>

B.1.1.10. Organization

Description	Value
Description:	If configured, all emails sent by the application will contain an X-Header with this organization or company name.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Organization'} = 'Example Company';</code>

B.1.1.11. DefaultCharset

Description	Value
Description:	Sets the default charset for the web interface to use (should represent the charset used to create the database or, in some cases, the database management system being used). "utf-8" is a good choice for environments expecting many charsets. You can specify another charset here (i.e. "iso-8859-1"). Please be sure that you will not be receiving foreign emails, or text, otherwise this could lead to problems.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'DefaultCharset'} = 'utf-8';</code>

B.1.1.12. DefaultLanguage

Description	Value
Description:	Defines the default front-end language. All the possible values are determined by the available language files on the system (see the next setting).
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'DefaultLanguage'} = 'en';</code>

B.1.1.13. DefaultUsedLanguages

Description	Value
Description:	Defines all the languages that are available to the application. The Key/Content pair links the front-end display name to the appropriate language PM file. The "Key" value should be the base-name of the PM file (i.e. de.pm is the file, then de is the "Key" value). The "Content" value should be the display name for the front-end. Specify any own-defined language here (see the developer documentation http://doc.ots.org/ for more information). Please remember to use the HTML equivalents for non-ASCII characters (i.e. for the German oe = o umlaut, it is necessary to use the ö symbol).
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1

[illegible]

B.1.1.14. DefaultTheme

Description	Value
Description:	Defines the default front-end (HTML) theme to be used by the agents and customers. The default themes are Standard and Lite. If you like, you can add your own theme. Please refer the administrator manual located at http://doc.otrs.org/ .
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'DefaultTheme'} = 'Standard';</code>

B.1.1.15. DefaultTheme::HostBased

Description	Value
Description:	It is possible to configure different themes, for example to distinguish between agents and customers, to be used on a per-domain basis within the application. Using a regular expression (regex), you can configure a Key/Content pair to match a domain. The value in "Key" should match the domain, and the value in "Content" should be a valid theme on your system. Please see the example entries for the proper form of the regex.
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'DefaultTheme::HostBased'} = { 'host1\\.example\\.com' => 'SomeTheme1', 'host2\\.example\\.com' => 'SomeTheme2' };</pre>

B.1.1.16. CheckMXRecord

Description	Value
Description:	Makes the application check the MX record of email addresses before sending an email or submitting a telephone or email ticket.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CheckMXRecord'} = '1';</pre>

B.1.1.17. CheckMXRecord::Nameserver

Description	Value
Description:	Defines the address of a dedicated DNS server, if necessary, for the "CheckMXRecord" look-ups.
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'CheckMXRecord::Nameserver'} = 'ns.example.com';</pre>

B.1.1.18. CheckEmailAddresses

Description	Value
Description:	Makes the application check the syntax of email addresses.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CheckEmailAddresses'} = '1';</code>

B.1.1.19. CheckEmailValidAddress

Description	Value
Description:	Defines a regular expression that excludes some addresses from the syntax check (if "CheckEmailAddresses" is set to "Yes"). Please enter a regex in this field for email addresses, that aren't syntactically valid, but are necessary for the system (i.e. "root@localhost").
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CheckEmailValidAddress'} = '^(root@localhost admin@localhost)';</code>

B.1.1.20. CheckEmailInvalidAddress

Description	Value
Description:	Defines a regular expression that filters all email addresses that should not be used in the application.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CheckEmailInvalidAddress'} = '@(example)\.\.(\. \.\.)*\$';</code>

B.1.1.21. CGILogPrefix

Description	Value
Description:	Specifies the text that should appear in the log file to denote a CGI script entry.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CGILogPrefix'} = 'OTRS-CGI';</code>

B.1.1.22. DemoSystem

Description	Value
Description:	Runs the system in "Demo" mode. If set to "Yes", agents can change preferences, such as selection of language and theme via the agent web interface. These changes are only valid for the current session. It will not be possible for agents to change their passwords.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'DemoSystem'} = '0';</code>

B.1.1.23. SwitchToUser

Description	Value
Description:	Allows the administrators to login as other users, via the users administration panel.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SwitchToUser'} = '0';</code>

B.1.1.24. NotificationSenderName

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

Description	Value
Description:	Specifies the name that should be used by the application when sending notifications. The sender name is used to build the complete display name for the notification master (i.e. "OTRS Notification Master" otrs@your.example.com). Notifications are messages such as en::Customer::QueueUpdate or en::Agent::Move.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'NotificationSenderName'} = 'OTRS Notification Master';</pre>

B.1.1.25. NotificationSenderEmail

Description	Value
Description:	Specifies the email address that should be used by the application when sending notifications. The email address is used to build the complete display name for the notification master (i.e. "OTRS Notification Master" otrs@your.example.com). You can use the OTRS_CONFIG_FQDN variable as set in your configuration, or choose another email address. Notifications are messages such as en::Customer::QueueUpdate or en::Agent::Move.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'NotificationSenderEmail'} = 'otrs@<OTRS_CONFIG_FQDN>';</pre>

B.1.1.26. System::Customer::Permission

Description	Value
Description:	Defines the standard permissions available for customers within the application. If more permissions are needed, you can enter them here. Permissions must be hard coded to be effective. Please ensure, when adding any of the afore mentioned permissions, that the "rw" permission remains the last entry.
Group:	Framework
SubGroup:	Core

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

B.1.1.27. LanguageDebug

Description	Value
Description:	Debugs the translation set. If this is set to "Yes" all strings (text) without translations are written to STDERR. This can be helpful when you are creating a new translation file. Otherwise, this option should remain set to "No".
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'LanguageDebug'} = '0';</code>

B.1.1.28. Secure::DisableBanner

Description	Value
Description:	If enabled, the OTRS version tag will be removed from the HTTP headers.
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Secure::DisableBanner'} = '0';</code>

B.1.2. Core::LinkObject

B.1.2.1. LinkObject::ViewMode

Description	Value
Description:	Determines the way the linked objects are displayed in each zoom mask.
Group:	Framework
SubGroup:	Core::LinkObject

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

B.1.2.2. LinkObject::Type###Normal

Description	Value
Description:	Defines the link type 'Normal'. If the source name and the target name contain the same value, the resulting link is a non-directional one; otherwise, the result is a directional link.
Group:	Framework
SubGroup:	Core::LinkObject
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LinkObject::Type'}->{'Normal'} = { 'SourceName' => 'Normal', 'TargetName' => 'Normal' };</code>

B.1.2.3. LinkObject::Type###ParentChild

Description	Value
Description:	Defines the link type 'ParentChild'. If the source name and the target name contain the same value, the resulting link is a non-directional one; otherwise, the result is a directional link.
Group:	Framework
SubGroup:	Core::LinkObject
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LinkObject::Type'}->{'ParentChild'} = { 'SourceName' => 'Parent', 'TargetName' => 'Child' };</code>

B.1.2.4. LinkObject::TypeGroup###0001

Description	Value
Description:	Defines the link type groups. The link types of the same group cancel one another. Example: If ticket A is linked per a 'Normal' link with ticket B, then these tickets could not be additionally linked with link of a 'ParentChild' relationship.

Description	Value
Group:	Framework
SubGroup:	Core::LinkObject
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'LinkObject::TypeGroup'}->{'0001'} = ['Normal', 'ParentChild'];</code>

B.1.3. Core::Log

B.1.3.1. LogModule

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

B.1.3.2. LogModule::SysLog::Facility

Description	Value
Description:	If "SysLog" was selected for LogModule, a special log facility can be specified.
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LogModule::SysLog::Facility'} = 'user';</code>

B.1.3.3. LogModule::SysLog::LogSock

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

Description	Value
Description:	If "SysLog" was selected for LogModule, a special log sock can be specified (on solaris you may need to use 'stream').
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LogModule::SysLog::LogSock'} = 'unix';</code>

B.1.3.4. LogModule::SysLog::Charset

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

B.1.3.5. LogModule::LogFile

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

B.1.3.6. LogModule::LogFile::Date

Description	Value
Description:	Adds a suffix with the actual year and month to the OTRS log file. A logfile for every month will be created.
Group:	Framework

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

B.1.4. Core::MIME-Viewer

B.1.4.1. MIME-Viewer###application/excel

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

B.1.4.2. MIME-Viewer###application/msword

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

B.1.4.3. MIME-Viewer###application/pdf

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

Description	Value
Description:	Specifies the path to the converter that allows the view of PDF documents, in the web interface.
Group:	Framework
SubGroup:	Core::MIME-Viewer
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'MIME-Viewer'}->{'application/pdf'} = 'pdftohtml -stdout -i';</code>

B.1.4.4. MIME-Viewer###text/xml

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

B.1.5. Core::MirrorDB

B.1.5.1. Core::MirrorDB::DSN

Description	Value
Description:	If you want to use a mirror database for agent ticket fulltext search or to generate stats, specify the DSN to this database.
Group:	Framework
SubGroup:	Core::MirrorDB
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Core::MirrorDB::DSN'} = 'DBI:mysql:database=mirrordb;host=mir</code>

B.1.5.2. Core::MirrorDB::User

Description	Value
Description:	If you want to use a mirror database for agent ticket fulltext search or to generate stats, the user to authenticate to this database can be specified.
Group:	Framework
SubGroup:	Core::MirrorDB
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Core::MirrorDB::User'} = 'some_user';</code>

B.1.5.3. Core::MirrorDB::Password

Description	Value
Description:	If you want to use a mirror database for agent ticket fulltext search or to generate stats, the password to authenticate to this database can be specified.
Group:	Framework
SubGroup:	Core::MirrorDB
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Core::MirrorDB::Password'} = 'some_password';</code>

B.1.6. Core::PDF**B.1.6.1. PDF**

Description	Value
Description:	Enables PDF output. The CPAN module PDF::API2 is required, if not installed, PDF output will be disabled.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PDF'} = '1';</code>

B.1.6.2. PDF::LogoFile

Description	Value
Description:	Specifies the path of the file for the logo in the page header (gif jpg png, 700 x 100 pixel).
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PDF::LogoFile'} = '<OTRS_CONFIG_Home>/var/logo-otrs.png';</code>

B.1.6.3. PDF::PageSize

Description	Value
Description:	Defines the standard size of PDF pages.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PDF::PageSize'} = 'a4';</code>

B.1.6.4. PDF::MaxPages

Description	Value
Description:	Defines the maximum number of pages per PDF file.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PDF::MaxPages'} = '100';</code>

B.1.6.5. PDF::TTFontFile###Proportional

Description	Value
Description:	Defines the path and TTF-File to handle proportional font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PDF::TTFontFile'}->{'Proportional'} = 'DejaVuSans.ttf';</code>

B.1.6.6. PDF::TTFontFile###ProportionalBold

Description	Value
Description:	Defines the path and TTF-File to handle bold proportional font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PDF::TTFontFile'}->{'ProportionalBold'} = 'DejaVuSans-Bold.ttf';</code>

B.1.6.7. PDF::TTFontFile###ProportionalItalic

Description	Value
Description:	Defines the path and TTF-File to handle italic proportional font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'PDF::TTFontFile'}->{'ProportionalItalic'} = 'DejaVuSans-Oblique.ttf';</code>

B.1.6.8. PDF::TTFontFile###ProportionalBoldItalic

Description	Value
Description:	Defines the path and TTF-File to handle bold italic proportional font in PDF documents.
Group:	Framework

Description	Value
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PDF::TTFontFile'}->{'ProportionalBoldItalic'} = 'DejaVuSans-B</code>

B.1.6.9. PDF::TTFontFile###Monospaced

Description	Value
Description:	Defines the path and TTF-File to handle monospaced font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PDF::TTFontFile'}->{'Monospaced'} = 'DejaVuSansMono.ttf';</code>

B.1.6.10. PDF::TTFontFile###MonospacedBold

Description	Value
Description:	Defines the path and TTF-File to handle bold monospaced font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'PDF::TTFontFile'}->{'MonospacedBold'} = 'DejaVuSansMono-Bold.</code>

B.1.6.11. PDF::TTFontFile###MonospacedItalic

Description	Value
Description:	Defines the path and TTF-File to handle italic monospaced font in PDF documents.

Description	Value
Group:	Framework
SubGroup:	Core::PDF
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'PDF::TTFontFile'}->{'MonospacedItalic'} = 'DejaVuSansMono-Obli'</code>

B.1.6.12. PDF::TTFontFile###MonospacedBoldItalic

Description	Value
Description:	Defines the path and TTF-File to handle bold italic monospaced font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'PDF::TTFontFile'}->{'MonospacedBoldItalic'} = 'DejaVuSansMono'</code>

B.1.7. Core::Package

B.1.7.1. Package::FileUpload

Description	Value
Description:	Enables file upload in the package manager frontend.
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Package::FileUpload'} = '1';</code>

B.1.7.2. Package::RepositoryRoot

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

Description	Value
Description:	Defines the location to get online repository list for additional packages. The first available result will be used.
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Package::RepositoryRoot'} = ['http://ftp.otrs.org/pub/otrs/misc/packages/repository.xml', 'http://otrs.org/repository.xml'];</pre>

B.1.7.3. Package::RepositoryList

Description	Value
Description:	Defines the list of online repositories. Another installations can be used as repository, for example: Key="http://example.com/otrs/public.pl?Action=PublicRepository;File=" and Content="Some Name".
Group:	Framework
SubGroup:	Core::Package
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Package::RepositoryList'} = { 'ftp://ftp.example.com/pub/otrs/misc/packages/' => '[Example] ftp://ftp</pre>

B.1.7.4. Package::RepositoryAccessRegExp

Description	Value
Description:	Defines the IP regular expression for accessing the local repository. You need to enable this to have access to your local repository and the package::RepositoryList is required on the remote host.
Group:	Framework
SubGroup:	Core::Package
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Package::RepositoryAccessRegExp'} = '127\.\0\.\0\.\1';</pre>

B.1.7.5. Package::Timeout

Description	Value
Description:	Sets the timeout (in seconds) for package downloads. Overwrites "WebUserAgent::Timeout".
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Package::Timeout'} = '15';</code>

B.1.7.6. Package::Proxy

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

B.1.8. Core::PerformanceLog**B.1.8.1. PerformanceLog**

Description	Value
Description:	Enables performance log (to log the page response time). It will affect the system performance. Frontend::Module###AdminPerformanceLog must be enabled.
Group:	Framework
SubGroup:	Core::PerformanceLog
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'PerformanceLog'} = '0';</code>

B.1.8.2. PerformanceLog::File

Description	Value
Description:	Specifies the path of the file for the performance log.
Group:	Framework
SubGroup:	Core::PerformanceLog
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PerformanceLog::File'} = '<OTRS_CONFIG_Home>/var/log/Performa</code>

B.1.8.3. PerformanceLog::FileMax

Description	Value
Description:	Defines the maximum size (in MB) of the log file.
Group:	Framework
SubGroup:	Core::PerformanceLog
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PerformanceLog::FileMax'} = '25';</code>

B.1.9. Core::SOAP**B.1.9.1. SOAP::User**

Description	Value
Description:	Defines the username to access the SOAP handle (bin/cgi-bin/rpc.pl).
Group:	Framework
SubGroup:	Core::SOAP
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'SOAP::User'} = 'some_user';</code>

B.1.9.2. SOAP::Password

Description	Value
Description:	Defines the password to access the SOAP handle (bin/cgi-bin/rpc.pl).

Description	Value
Group:	Framework
SubGroup:	Core::SOAP
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'SOAP::Password'} = 'some_pass';</code>

B.1.10. Core::Sendmail

B.1.10.1. SendmailModule

Description	Value
Description:	Defines the module to send emails. "Sendmail" directly uses the sendmail binary of your operating system. Any of the "SMTP" mechanisms use a specified (external) mailserver. "DoNotSendEmail" doesn't send emails and it is useful for test systems.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SendmailModule'} = 'Kernel::System::Email::Sendmail';</code>

B.1.10.2. SendmailModule::CMD

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

B.1.10.3. SendmailModule::Host

Description	Value
Description:	If any of the "SMTP" mechanisms was selected as SendmailModule, the mailhost that sends out the mails must be specified.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SendmailModule::Host'} = 'mail.example.com';</code>

B.1.10.4. SendmailModule::Port

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

B.1.10.5. SendmailModule::AuthUser

Description	Value
Description:	If any of the "SMTP" mechanisms was selected as SendmailModule, and authentication to the mail server is needed, an username must be specified.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'SendmailModule::AuthUser'} = 'MailserverLogin';</code>

B.1.10.6. SendmailModule::AuthPassword

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

Description	Value
Description:	If any of the "SMTP" mechanisms was selected as SendmailModule, and authentication to the mail server is needed, a password must be specified.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'SendmailModule::AuthPassword'} = 'MailserverPassword';</code>

B.1.10.7. SendmailBcc

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

B.1.10.8. SendmailNotificationEnvelopeFrom

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

B.1.10.9. SendmailEncodingForce

Description	Value
Description:	Forces encoding of outgoing emails (7bit 8bit quoted-printable base64).
Group:	Framework
SubGroup:	Core::Sendmail

Description	Value
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'SendmailEncodingForce'} = 'base64';</code>

B.1.11. Core::Session

B.1.11.1. SessionModule

Description	Value
Description:	Defines the module used to store the session data. With "DB" the frontend server can be splitted from the db server. "FS" is faster.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionModule'} = 'Kernel::System::AuthSession::DB';</code>

B.1.11.2. SessionName

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

B.1.11.3. SessionCheckRemoteIP

Description	Value
Description:	Turns on the remote ip address check. It should be set to "No" if the application is used, for example, via a proxy farm or a dialup connection, because the remote ip address is mostly different for the requests.
Group:	Framework
SubGroup:	Core::Session

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

B.1.11.4. SessionDeleteIfNotRemoteID

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

B.1.11.5. SessionMaxTime

Description	Value
Description:	Defines the maximal valid time (in seconds) for a session id.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionMaxTime'} = '57600';</code>

B.1.11.6. SessionMaxIdleTime

Description	Value
Description:	Sets the inactivity time (in seconds) to pass before a session is killed and a user is logged out.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionMaxIdleTime'} = '21600';</code>

B.1.11.7. SessionDeleteIfTimeToOld

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

B.1.11.8. SessionUseCookie

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

B.1.11.9. SessionUseCookieAfterBrowserClose

Description	Value
Description:	Stores cookies after the browser has been closed.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionUseCookieAfterBrowserClose'} = '0';</code>

B.1.11.10. SessionCSRFProtection

Description	Value
Description:	Protection against CSRF (Cross Site Request Forgery) exploits (for more info see http://en.wikipedia.org/wiki/Cross-site_request_forgery).
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1

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

B.1.11.11. SessionDir

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

B.1.11.12. SessionTable

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

B.1.11.13. SessionTableID

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

B.1.11.14. SessionTableValue

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

B.1.11.15. CustomerPanelSessionName

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

B.1.12. Core::SpellChecker**B.1.12.1. SpellChecker**

Description	Value
Description:	Enables spell checker support.
Group:	Framework
SubGroup:	Core::SpellChecker
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SpellChecker'} = '0';</code>

B.1.12.2. SpellCheckerBin

Description	Value
Description:	Install ispell or aspell on the system, if you want to use a spell checker. Please specify the path to the aspell or ispell binary on your operating system.

Description	Value
Group:	Framework
SubGroup:	Core::SpellChecker
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SpellCheckerBin'} = '/usr/bin/ispell';</code>

B.1.12.3. SpellCheckerDictDefault

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

B.1.12.4. SpellCheckerIgnore

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

B.1.13. Core::Stats

B.1.13.1. Stats::StatsHook

Description	Value
Description:	Sets the stats hook.
Group:	Framework
SubGroup:	Core::Stats
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'Stats::StatsHook'} = 'Stat#';</code>

B.1.13.2. Stats::StatsStartNumber

Description	Value
Description:	Start number for statistics counting. Every new stat increments this number.
Group:	Framework
SubGroup:	Core::Stats
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::StatsStartNumber'} = '10000';</code>

B.1.13.3. Stats::MaxXaxisAttributes

Description	Value
Description:	Defines the default maximum number of X-axis attributes for the time scale.
Group:	Framework
SubGroup:	Core::Stats
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Stats::MaxXaxisAttributes'} = '1000';</code>

B.1.14. Core::Stats::Graph

B.1.14.1. Stats::Graph::t_margin

Description	Value
Description:	Specifies the top margin of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::t_margin'} = '10';</code>

B.1.14.2. Stats::Graph::l_margin

Description	Value
Description:	Specifies the left margin of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::l_margin'} = '10';</code>

B.1.14.3. Stats::Graph::b_margin

Description	Value
Description:	Specifies the bottom margin of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::b_margin'} = '10';</code>

B.1.14.4. Stats::Graph::r_margin

Description	Value
Description:	Specifies the right margin of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::r_margin'} = '20';</code>

B.1.14.5. Stats::Graph::bgclr

Description	Value
Description:	Specifies the background color of the picture.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Stats::Graph::bgclr'} = 'white';</code>

B.1.14.6. Stats::Graph::transparent

Description	Value
Description:	Makes the picture transparent.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::transparent'} = '0';</code>

B.1.14.7. Stats::Graph::fgclr

Description	Value
Description:	Specifies the border color of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::fgclr'} = 'black';</code>

B.1.14.8. Stats::Graph::boxclr

Description	Value
Description:	Specifies the background color of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::boxclr'} = 'white';</code>

B.1.14.9. Stats::Graph::accentclr

Description	Value
Description:	Specifies the border color of the legend.

Description	Value
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::accentclr'} = 'black';</code>

B.1.14.10. Stats::Graph::legendclr

Description	Value
Description:	Specifies the text color of the legend.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::legendclr'} = 'black';</code>

B.1.14.11. Stats::Graph::textclr

Description	Value
Description:	Specifies the text color of the chart (e. g. caption).
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::textclr'} = 'black';</code>

B.1.14.12. Stats::Graph::dclrs

Description	Value
Description:	Defines the colors for the graphs.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self->{'Stats::Graph::dclrs'} = ['red', 'green', 'blue', 'yellow', 'black', 'purple', 'orange', 'pink', 'marine', 'cyan', 'lgray', 'lblue', 'lyellow', 'lgreen', 'lred', 'lpurple', 'lorange', 'lbrown'];</pre>

B.1.14.13. Stats::Graph::line_width

Description	Value
Description:	Defines the boldness of the line drawn by the graph.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Stats::Graph::line_width'} = '1';</pre>

B.1.14.14. Stats::Graph::legend_placement

Description	Value
Description:	Defines the placement of the legend. This should be a two letter key of the form: 'B[LCR] R[TCB]'. The first letter indicates the placement (Bottom or Right), and the second letter the alignment (Left, Right, Center, Top, or Bottom).
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Stats::Graph::legend_placement'} = 'BC';</pre>

B.1.14.15. Stats::Graph::legend_spacing

Description	Value
Description:	Defines the spacing of the legends.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Stats::Graph::legend_spacing'} = '4';</pre>

B.1.14.16. Stats::Graph::legend_marker_width

Description	Value
Description:	Defines the width of the legend.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::legend_marker_width'} = '12';</code>

B.1.14.17. Stats::Graph::legend_marker_height

Description	Value
Description:	Defines the height of the legend.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::legend_marker_height'} = '8';</code>

B.1.15. Core::Time**B.1.15.1. TimeInputFormat**

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

B.1.15.2. TimeShowAlwaysLong

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

Description	Value
Description:	Shows time in long format (days, hours, minutes), if set to "Yes"; or in short format (days, hours), if set to "No".
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeShowAlwaysLong'} = '0';</code>

B.1.15.3. TimeZone

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

B.1.15.4. TimeZoneUser

Description	Value
Description:	Sets the user time zone per user (required a system with UTC as system time and UTC under TimeZone). Otherwise this is a diff time to the local time.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'TimeZoneUser'} = '0';</code>

B.1.15.5. TimeZoneUserBrowserAutoOffset

Description	Value
Description:	Sets the user time zone per user based on java script / browser time zone offset feature at login time.
Group:	Framework
SubGroup:	Core::Time
Valid:	1

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

B.1.15.6. CalendarWeekDayStart

Description	Value
Description:	Define the start day of the week for the date picker.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CalendarWeekDayStart'} = '1';</code>

B.1.15.7. TimeVacationDays

Description	Value
Description:	Adds the permanent vacation days. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDays'} = { '1' => { '1' => 'New Year\'s Day' }, '12' => { '24' => 'Christmas Eve', '25' => 'First Christmas Day', '26' => 'Second Christmas Day', '31' => 'New Year\'s Eve' }, '5' => { '1' => 'International Workers\' Day' } };</code>

B.1.15.8. TimeVacationDaysOneTime

Description	Value
Description:	Adds the one time vacation days. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDaysOneTime'} = { '2004' => { '1' => { '1' => 'test' } } };</code>

B.1.15.9. TimeWorkingHours

Description	Value
Description:	Defines the hours and week days to count the working time.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeWorkingHours'} = { 'Fri' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Mon' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Sat' => [], 'Sun' => [], 'Thu' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Tue' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Wed' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'] };</code>

B.1.15.10. TimeShowCompleteDescription

Description	Value
Description:	Shows time use complete description (days, hours, minutes), if set to "Yes"; or just first letter (d, h, m), if set to "No".
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeShowCompleteDescription'} = '0';</code>

B.1.16. Core::Time::Calendar1

B.1.16.1. TimeZone::Calendar1Name

Description	Value
Description:	Defines the name of the calendar number 1.
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeZone::Calendar1Name'} = 'Calendar Name 1';</code>

B.1.16.2. TimeZone::Calendar1

Description	Value
Description:	Defines the time zone of the calendar number 1, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeZone::Calendar1'} = '0';</code>

B.1.16.3. TimeVacationDays::Calendar1

Description	Value
Description:	Adds the permanent vacation days for the calendar number 1. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self->{'TimeVacationDays::Calendar1'} = { '1' => 'New Year\'s Day' }, '12' => { '24' => 'Christmas Eve', '25' => 'First Christmas Day', '26' => 'Second Christmas Day', '31' => 'New Year\'s Eve' }, '5' => { '1' => 'International Workers\' Day' } };</pre>

B.1.16.4. TimeVacationDaysOneTime::Calendar1

Description	Value
Description:	Adds the one time vacation days for the calendar number 1. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeVacationDaysOneTime::Calendar1'} = { '2004' => { '1' => { '1' => 'test' } } };</pre>

B.1.16.5. TimeWorkingHours::Calendar1

Description	Value
Description:	Defines the hours and week days of the calendar number 1, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self->{'TimeWorkingHours::Calendar1'} = { 'Fri' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Mon' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Sat' => [], 'Sun' => [], 'Thu' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Tue' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Wed' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'] }; </pre>

B.1.17. Core::Time::Calendar2

B.1.17.1. TimeZone::Calendar2Name

Description	Value
Description:	Defines the name of the calendar number 2.
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeZone::Calendar2Name'} = 'Calendar Name 2'; </pre>

B.1.17.2. TimeZone::Calendar2

Description	Value
Description:	Defines the time zone of the calendar number 2, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'TimeZone::Calendar2'} = '0';</code>

B.1.17.3. TimeVacationDays::Calendar2

Description	Value
Description:	Adds the permanent vacation days for the calendar number 2. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDays::Calendar2'} = { '1' => { '1' => 'New Year\'s Day' }, '12' => { '24' => 'Christmas Eve', '25' => 'First Christmas Day', '26' => 'Second Christmas Day', '31' => 'New Year\'s Eve' }, '5' => { '1' => 'International Workers\' Day' } };</code>

B.1.17.4. TimeVacationDaysOneTime::Calendar2

Description	Value
Description:	Adds the one time vacation days for the calendar number 2. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDaysOneTime::Calendar2'} = { '2004' => { '1' => { '1' => 'test' } } };</code>

B.1.17.5. TimeWorkingHours::Calendar2

Description	Value
Description:	Defines the hours and week days of the calendar number 2, to count the working time.
Group:	Framework

Description	Value
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeWorkingHours::Calendar2'} = { 'Fri' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Mon' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Sat' => [], 'Sun' => [], 'Thu' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Tue' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Wed' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'] };</pre>

B.1.18. Core::Time::Calendar3

B.1.18.1. TimeZone::Calendar3Name

Description	Value
Description:	Defines the name of the calendar number 3.
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeZone::Calendar3Name'} = 'Calendar Name 3';</pre>

B.1.18.2. TimeZone::Calendar3

Description	Value
Description:	Defines the time zone of the calendar number 3, which can be assigned later to a specific queue.
Group:	Framework

Description	Value
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeZone::Calendar3'} = '0';</code>

B.1.18.3. TimeVacationDays::Calendar3

Description	Value
Description:	Adds the permanent vacation days for the calendar number 3. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDays::Calendar3'} = { '1' => { '1' => 'New Year\'s Day' }, '12' => { '24' => 'Christmas Eve', '25' => 'First Christmas Day', '26' => 'Second Christmas Day', '31' => 'New Year\'s Eve' }, '5' => { '1' => 'International Workers\' Day' } };</code>

B.1.18.4. TimeVacationDaysOneTime::Calendar3

Description	Value
Description:	Adds the one time vacation days for the calendar number 3. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDaysOneTime::Calendar3'} = { '2004' => { '1' => { '1' => 'test' } } };</code>

B.1.18.5. TimeWorkingHours::Calendar3

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

Description	Value
Description:	Defines the hours and week days of the calendar number 3, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeWorkingHours::Calendar3'} = { 'Fri' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Mon' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Sat' => [], 'Sun' => [], 'Thu' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Tue' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Wed' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'] }; </pre>

B.1.19. Core::Time::Calendar4

B.1.19.1. TimeZone::Calendar4Name

Description	Value
Description:	Defines the name of the calendar number 4.
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeZone::Calendar4Name'} = 'Calendar Name 4'; </pre>

B.1.19.2. TimeZone::Calendar4

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

Description	Value
Description:	Defines the time zone of the calendar number 4, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeZone::Calendar4'} = '0';</code>

B.1.19.3. TimeVacationDays::Calendar4

Description	Value
Description:	Adds the permanent vacation days for the calendar number 4. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDays::Calendar4'} = { '1' => { '1' => 'New Year\'s Day' }, '12' => { '24' => 'Christmas Eve', '25' => 'First Christmas Day', '26' => 'Second Christmas Day', '31' => 'New Year\'s Eve' }, '5' => { '1' => 'International Workers\' Day' } };</code>

B.1.19.4. TimeVacationDaysOneTime::Calendar4

Description	Value
Description:	Adds the one time vacation days for the calendar number 4. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDaysOneTime::Calendar4'} = { '2004' => { '1' => { '1' => 'test' } } };</code>

B.1.19.5. TimeWorkingHours::Calendar4

Description	Value
Description:	Defines the hours and week days of the calendar number 4, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeWorkingHours::Calendar4'} = { 'Fri' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Mon' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Sat' => [], 'Sun' => [], 'Thu' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Tue' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Wed' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'] }; </pre>

B.1.20. Core::Time::Calendar5**B.1.20.1. TimeZone::Calendar5Name**

Description	Value
Description:	Defines the name of the calendar number 5.
Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeZone::Calendar5Name'} = 'Calendar Name 5'; </pre>

B.1.20.2. TimeZone::Calendar5

Description	Value
Description:	Defines the time zone of the calendar number 5, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeZone::Calendar5'} = '0';</code>

B.1.20.3. TimeVacationDays::Calendar5

Description	Value
Description:	Adds the permanent vacation days for the calendar number 5. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDays::Calendar5'} = { '1' => { '1' => 'New Year\'s Day' }, '12' => { '24' => 'Christmas Eve', '25' => 'First Christmas Day', '26' => 'Second Christmas Day', '31' => 'New Year\'s Eve' }, '5' => { '1' => 'International Workers\' Day' } };</code>

B.1.20.4. TimeVacationDaysOneTime::Calendar5

Description	Value
Description:	Adds the one time vacation days for the calendar number 5. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDaysOneTime::Calendar5'} = { '2004' => { '1' => { '1' => 'test' } } };</code>

B.1.20.5. TimeWorkingHours::Calendar5

Description	Value
Description:	Defines the hours and week days of the calendar number 5, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeWorkingHours::Calendar5'} = { 'Fri' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Mon' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Sat' => [], 'Sun' => [], 'Thu' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Tue' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Wed' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'] }; </pre>

B.1.21. Core::Time::Calendar6**B.1.21.1. TimeZone::Calendar6Name**

Description	Value
Description:	Defines the name of the calendar number 6.
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeZone::Calendar6Name'} = 'Calendar Name 6'; </pre>

B.1.21.2. TimeZone::Calendar6

Description	Value
Description:	Defines the time zone of the calendar number 6, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeZone::Calendar6'} = '0';</code>

B.1.21.3. TimeVacationDays::Calendar6

Description	Value
Description:	Adds the permanent vacation days for the calendar number 6. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDays::Calendar6'} = { '1' => { '1' => 'New Year\'s Day' }, '12' => { '24' => 'Christmas Eve', '25' => 'First Christmas Day', '26' => 'Second Christmas Day', '31' => 'New Year\'s Eve' }, '5' => { '1' => 'International Workers\' Day' } };</code>

B.1.21.4. TimeVacationDaysOneTime::Calendar6

Description	Value
Description:	Adds the one time vacation days for the calendar number 6. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDaysOneTime::Calendar6'} = { '2004' => { '1' => { '1' => 'test' } } };</code>

B.1.21.5. TimeWorkingHours::Calendar6

Description	Value
Description:	Defines the hours and week days of the calendar number 6, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeWorkingHours::Calendar6'} = { 'Fri' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Mon' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Sat' => [], 'Sun' => [], 'Thu' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Tue' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Wed' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'] }; </pre>

B.1.22. Core::Time::Calendar7**B.1.22.1. TimeZone::Calendar7Name**

Description	Value
Description:	Defines the name of the calendar number 7.
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeZone::Calendar7Name'} = 'Calendar Name 7'; </pre>

B.1.22.2. TimeZone::Calendar7

Description	Value
Description:	Defines the time zone of the calendar number 7, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeZone::Calendar7'} = '0';</code>

B.1.22.3. TimeVacationDays::Calendar7

Description	Value
Description:	Adds the permanent vacation days for the calendar number 7. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDays::Calendar7'} = { '1' => { '1' => 'New Year\'s Day' }, '12' => { '24' => 'Christmas Eve', '25' => 'First Christmas Day', '26' => 'Second Christmas Day', '31' => 'New Year\'s Eve' }, '5' => { '1' => 'International Workers\' Day' } };</code>

B.1.22.4. TimeVacationDaysOneTime::Calendar7

Description	Value
Description:	Adds the one time vacation days for the calendar number 7. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDaysOneTime::Calendar7'} = { '2004' => { '1' => { '1' => 'test' } } };</code>

B.1.22.5. TimeWorkingHours::Calendar7

Description	Value
Description:	Defines the hours and week days of the calendar number 7, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeWorkingHours::Calendar7'} = { 'Fri' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Mon' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Sat' => [], 'Sun' => [], 'Thu' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Tue' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Wed' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'] }; </pre>

B.1.23. Core::Time::Calendar8**B.1.23.1. TimeZone::Calendar8Name**

Description	Value
Description:	Defines the name of the calendar number 8.
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeZone::Calendar8Name'} = 'Calendar Name 8'; </pre>

B.1.23.2. TimeZone::Calendar8

Description	Value
Description:	Defines the time zone of the calendar number 8, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeZone::Calendar8'} = '0';</code>

B.1.23.3. TimeVacationDays::Calendar8

Description	Value
Description:	Adds the permanent vacation days for the calendar number 8. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDays::Calendar8'} = { '1' => { '1' => 'New Year\'s Day' }, '12' => { '24' => 'Christmas Eve', '25' => 'First Christmas Day', '26' => 'Second Christmas Day', '31' => 'New Year\'s Eve' }, '5' => { '1' => 'International Workers\' Day' } };</code>

B.1.23.4. TimeVacationDaysOneTime::Calendar8

Description	Value
Description:	Adds the one time vacation days for the calendar number 8. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDaysOneTime::Calendar8'} = { '2004' => { '1' => { '1' => 'test' } } };</code>

B.1.23.5. TimeWorkingHours::Calendar8

Description	Value
Description:	Defines the hours and week days of the calendar number 8, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeWorkingHours::Calendar8'} = { 'Fri' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Mon' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Sat' => [], 'Sun' => [], 'Thu' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Tue' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Wed' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'] }; </pre>

B.1.24. Core::Time::Calendar9**B.1.24.1. TimeZone::Calendar9Name**

Description	Value
Description:	Defines the name of the calendar number 9.
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeZone::Calendar9Name'} = 'Calendar Name 9'; </pre>

B.1.24.2. TimeZone::Calendar9

Description	Value
Description:	Defines the time zone of the calendar number 9, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeZone::Calendar9'} = '0';</code>

B.1.24.3. TimeVacationDays::Calendar9

Description	Value
Description:	Adds the permanent vacation days for the calendar number 9. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDays::Calendar9'} = { '1' => { '1' => 'New Year\'s Day' }, '12' => { '24' => 'Christmas Eve', '25' => 'First Christmas Day', '26' => 'Second Christmas Day', '31' => 'New Year\'s Eve' }, '5' => { '1' => 'International Workers\' Day' } };</code>

B.1.24.4. TimeVacationDaysOneTime::Calendar9

Description	Value
Description:	Adds the one time vacation days for the calendar number 9. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDaysOneTime::Calendar9'} = { '2004' => { '1' => { '1' => 'test' } } };</code>

B.1.24.5. TimeWorkingHours::Calendar9

Description	Value
Description:	Defines the hours and week days of the calendar number 9, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeWorkingHours::Calendar9'} = { 'Fri' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Mon' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Sat' => [], 'Sun' => [], 'Thu' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Tue' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Wed' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'] }; </pre>

B.1.25. Core::Web**B.1.25.1. Frontend::WebPath**

Description	Value
Description:	Defines the URL base path of icons, CSS and Java Script.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Frontend::WebPath'} = '/otrs-web/'; </pre>

B.1.25.2. Frontend::ImagePath

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

Description	Value
Description:	Defines the URL image path of icons for navigation.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::ImagePath'} = '<OTRS_CONFIG_Frontend::WebPath>skins</code>

B.1.25.3. Frontend::CSSPath

Description	Value
Description:	Defines the URL CSS path.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::CSSPath'} = '<OTRS_CONFIG_Frontend::WebPath>css/';</code>

B.1.25.4. Frontend::JavaScriptPath

Description	Value
Description:	Defines the URL java script path.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::JavaScriptPath'} = '<OTRS_CONFIG_Frontend::WebPath></code>

B.1.25.5. Frontend::RichText

Description	Value
Description:	Uses richtext for viewing and editing: articles, salutations, signatures, standard responses, auto responses and notifications.

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

B.1.25.6. Frontend::RichTextPath

Description	Value
Description:	Defines the URL rich text editor path.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::RichTextPath'} = '<OTRS_CONFIG_Frontend::WebPath>js</code>

B.1.25.7. Frontend::RichTextWidth

Description	Value
Description:	Defines the width for the rich text editor component. Enter number (pixels) or percent value (relative).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::RichTextWidth'} = '620';</code>

B.1.25.8. Frontend::RichTextHeight

Description	Value
Description:	Defines the =hHeight for the rich text editor component. Enter number (pixels) or percent value (relative).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Frontend::RichTextHeight'} = '320';</code>

B.1.25.9. Frontend::RichText::DefaultCSS

Description	Value
Description:	Defines the default CSS used in rich text editors.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::RichText::DefaultCSS'} = 'font-family: Geneva, Helvet</code>

B.1.25.10. DefaultViewNewLine

Description	Value
Description:	Automated line break in text messages after x number of chars.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'DefaultViewNewLine'} = '90';</code>

B.1.25.11. DefaultPreViewLines

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

B.1.25.12. DefaultViewLines

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

B.1.25.13. Frontend::AnimationEnabled

Description	Value
Description:	Turns on the animations used in the GUI. If you have problems with these animations (e.g. performance issues), you can turn them off here.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::AnimationEnabled'} = '1';</code>

B.1.25.14. AttachmentDownloadType

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

B.1.25.15. WebMaxFileUpload

Description	Value
Description:	Defines the maximal size (in bytes) for file uploads via the browser.
Group:	Framework
SubGroup:	Core::Web
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'WebMaxFileUpload'} = '16777216';</code>

B.1.25.16. WebUploadCacheModule

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

B.1.25.17. Frontend::Output::FilterText###AAAURL

Description	Value
Description:	Defines the filter that processes the text in the articles, in order to highlight URLs.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Frontend::Output::FilterText'}->{'AAAURL'} = { 'Module' => 'Kernel::Output::HTML::OutputFilterTextURL' };</code>

B.1.25.18. Frontend::Themes

Description	Value
Description:	Activates the available themes on the system. Value 1 means active, 0 means inactive.
Group:	Framework
SubGroup:	Core::Web
Valid:	1

Description	Value
Required:	0
Config-Setting:	<code>\$Self->{'Frontend::Themes'} = { 'Lite' => '0', 'Standard' => '1' };</code>

B.1.25.19. Frontend::Output::FilterText###OutputFilterTextAutoLink

Description	Value
Description:	Defines a filter to process the text in the articles, in order to highlight predefined keywords.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Frontend::Output::FilterText'}->{'OutputFilterTextAutoLink'} = 'Module' => 'Kernel::Output::HTML::OutputFilterTextAutoLink' };</code>

B.1.25.20. Frontend::Output::OutputFilterTextAutoLink###CVE

Description	Value
Description:	Defines a filter for html output to add links behind CVE numbers. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possibility is to insert the link to the image.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Output::OutputFilterTextAutoLink'}->{'CVE'} = { 'RegExp' => ['(CVE CAN)\\-(\\d{3,4})\\-(\\d{2,6})'], 'URL1' => { 'Description' => 'Mitre', 'Image' => 'http://cve.mitre.org/favicon.ico', 'Target' => '_blank', 'URL' => 'http://cve.mitre.org/cgi-bin/cvename.cgi?name=<MATCH1>-<MATCH2>', }, 'URL2' => { 'Description' => 'Google', 'Image' => 'http://www.google.de/favicon.ico', 'Target' => '_blank', 'URL' => 'http://google.com/search?q=<MATCH1>-<MATCH2>-<MATCH3>', }, 'URL3' => { 'Description' => 'US-CERT NVD', 'Image' => 'http://nvd.nist.gov/favicon.ico', 'Target' => '_blank', 'URL' => 'http://nvd.nist.gov/nvd.cfm?cvename=<MATCH1>-<MATCH2>-<MATCH3>', } }; </pre>

B.1.25.21. Frontend::Output::OutputFilterTextAutoLink###Bugtraq

Description	Value
Description:	Defines a filter for html output to add links behind bugtraq numbers. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possibility is to insert the link to the image.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Output::OutputFilterTextAutoLink'}->{'Bugtraq'} = { 'RegExp' => ['Bugtraq[\\s\\w\\t]*?ID[\\s\\w\\t]*?:[\\s\\w\\t]*?(\\d{2,8})', 'Bugtraq[\\s\\w\\t]*?ID[\\s\\w\\t]*?(\\d{2,8})', 'Bugtraq[\\s\\w\\t]*?:[\\s\\w\\t]*?(\\d{2,8})', 'Bugtraq[\\s\\w\\t]*?(\\d{2,8})', 'BID[\\s\\w\\t]*?:[\\s\\w\\t]*?(\\d{2,8})', 'BID[\\s\\w\\t]*?(\\d{2,8})'], 'URL1' => { 'Description' => 'Security Focus', 'Image' => 'http://www.securityfocus.com/favicon.ico', 'Target' => '_blank', 'URL' => 'http://www.securityfocus.com/bid/<MATCH1>/info' }, 'URL2' => { 'Description' => 'Google', 'Image' => 'http://www.google.de/favicon.ico', 'Target' => '_blank', 'URL' => 'http://google.com/search?q=<MATCH>' } }; </pre>

B.1.25.22. Frontend::Output::OutputFilterTextAutoLink###MSBulletins

Description	Value
Description:	Defines a filter for html output to add links behind MSBulletin numbers. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possibility is to insert the link to the image.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Output::OutputFilterTextAutoLink'}->{'MSBulletins'} = { 'RegExp' => ['MS^[A-Za-z]{0,5}(\\d\\d).?(\\d{2,4})'], 'URL1' => { 'Description' => 'Microsoft Technet', 'Image' => 'http://www.microsoft.com/favicon.ico', 'Target' => '_blank', 'URL' => 'http://www.microsoft.com/technet/security/bulletin/MS<MATCH1>' }, 'URL2' => { 'Description' => 'Google', 'Image' => 'http://www.google.de/favicon.ico', 'Target' => '_blank', 'URL' => 'http://google.com/search?q=MS<MATCH1>-<MATCH2>' } }; </pre>

B.1.25.23. Frontend::Output::OutputFilterTextAutoLink###Setting1

Description	Value
Description:	Define a filter for html output to add links behind a defined string. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possibility is to insert the link to the image.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Output::OutputFilterTextAutoLink'}->{'Setting1'} = 'RegExp' => ['RegExp'], 'URL1' => { 'Description' => 'Description', 'Image' => 'right-small.png', 'Target' => '_blank', 'URL' => 'URL' }, 'URL2' => { 'Description' => 'Description', 'Image' => 'Image', 'Target' => '_blank', 'URL' => 'URL' } };</pre>

B.1.25.24. Frontend::Output::OutputFilterTextAutoLink###Setting2

Description	Value
Description:	Defines a filter for html output to add links behind a defined string. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possibility is to insert the link to the image.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Output::OutputFilterTextAutoLink'}->{'Setting2'} = 'RegExp' => ['RegExp'], 'URL1' => { 'Description' => 'Description', 'Image' => 'right-small.png', 'Target' => '_blank', 'URL' => 'URL' }, 'URL2' => { 'Description' => 'Description', 'Image' => 'Image', 'Target' => '_blank', 'URL' => 'URL' }, 'URL3' => { 'Description' => 'Description', 'Image' => 'Image', 'Target' => '_blank', 'URL' => 'URL' } };</pre>

B.1.25.25. Loader::Enabled::CSS

Description	Value
Description:	If enabled, OTRS will deliver all CSS files in minified form. WARNING: If you turn this off, there will likely be problems in IE 7, because it cannot load more than 32 CSS files.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Loader::Enabled::CSS'} = '1';</code>

B.1.25.26. Loader::Enabled::JS

Description	Value
Description:	If enabled, OTRS will deliver all JavaScript files in minified form.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Loader::Enabled::JS'} = '1';</code>

B.1.25.27. Loader::Agent::CommonCSS###000-Framework

Description	Value
Description:	List of CSS files to always be loaded for the agent interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'Loader::Agent::CommonCSS'}->{'000-Framework'} = ['Core.Reset.css', 'Core.Default.css', 'Core.Header.css', 'Core.OverviewControl.css', 'Core.OverviewSmall.css', 'Core.OverviewMedium.css', 'Core.OverviewLarge.css', 'Core.Footer.css', 'Core.PageLayout.css', 'Core.Form.css', 'Core.Table.css', 'Core.Widget.css', 'Core.WidgetMenu.css', 'Core.TicketDetail.css', 'Core.Tooltip.css', 'Core.Dialog.css', 'Core.Print.css']; </pre>

B.1.25.28. Loader::Agent::CommonCSS::IE7###000-Framework

Description	Value
Description:	List of IE7-specific CSS files to always be loaded for the agent interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Loader::Agent::CommonCSS::IE7'}->{'000-Framework'} = ['Core.Default.IE7.css', 'Core.Header.IE7.css', 'Core.OverviewControl.IE7.css', 'Core.OverviewSmall.IE7.css', 'Core.OverviewMedium.IE7.css', 'Core.OverviewLarge.IE7.css', 'Core.PageLayout.IE7.css', 'Core.Form.IE7.css', 'Core.Widget.IE7.css', 'Core.WidgetMenu.IE7.css', 'Core.Tooltip.IE7.css', 'Core.Dialog.IE7.css', 'Core.TicketDetail.IE7.css', 'Core.Table.IE7.css'];</pre>

B.1.25.29. Loader::Agent::CommonCSS::IE8###000-Framework

Description	Value
Description:	List of IE8-specific CSS files to always be loaded for the agent interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Loader::Agent::CommonCSS::IE8'}->{'000-Framework'} = ['Core.OverviewSmall.IE8.css'];</pre>

B.1.25.30. Loader::Agent::CommonJS###000-Framework

Description	Value
Description:	List of JS files to always be loaded for the agent interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre> \$Self->{'Loader::Agent::CommonJS'}->{'000-Framework'} = ['thirdparty/json/json2.js', 'thirdparty/jquery-1.4.4/jquery.js', 'thirdparty/jquery-ui-1.8.7/jquery-ui.js', 'thirdparty/jquery-validate-1.7/jquery.validate.js', 'thirdparty/stacktrace/stacktrace.js', 'Core.JavaScriptEnhancements.js', 'Core.Debug.js', 'Core.Data.js', 'Core.Config.js', 'Core.Exception.js', 'Core.JSON.js', 'Core.AJAX.js', 'Core.UI.js', 'Core.UI.IE7Fixes.js', 'Core.UI.Accordion.js', 'Core.UI.Datepicker.js', 'Core.UI.Resizable.js', 'Core.UI.Table.js', 'Core.UI.Accessibility.js', 'Core.UI.RichTextEditor.js', 'Core.UI.Dialog.js', 'Core.UI.ActionRow.js', 'Core.UI.Popup.js', 'Core.Form.js', 'Core.Form.ErrorToolTips.js', 'Core.Form.Validate.js', 'Core.App.js', 'Core.Agent.js', 'Core.Agent.Search.js']; </pre>

B.1.25.31. Loader::Customer::CommonCSS###000-Framework

Description	Value
Description:	List of CSS files to always be loaded for the customer interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'Loader::Customer::CommonCSS'}->{'000-Framework'} = ['Core.Reset.css', 'Core.Default.css', 'Core.Form.css', 'Core.Dialog.css', 'Core.Tooltip.css', 'Core.Login.css', 'Core.Control.css', 'Core.Table.css', 'Core.TicketZoom.css', 'Core.Print.css']; </pre>

B.1.25.32. Loader::Customer::CommonCSS::IE6###000-Framework

Description	Value
Description:	List of IE6-specific CSS files to always be loaded for the customer interface.
Group:	Framework
SubGroup:	Core::Web

Description	Value
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Loader::Customer::CommonCSS::IE6'}->{'000-Framework'} = ['Core.IE6.css', 'Core.Tooltip.IE6.css'];</pre>

B.1.25.33. Loader::Customer::CommonCSS::IE7###000-Framework

Description	Value
Description:	List of IE7-specific CSS files to always be loaded for the customer interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Loader::Customer::CommonCSS::IE7'}->{'000-Framework'} = ['Core.IE7.css', 'Core.Tooltip.IE7.css', 'Core.Dialog.IE7.css'];</pre>

B.1.25.34. Loader::Customer::CommonCSS::IE8###000-Framework

Description	Value
Description:	List of IE8-specific CSS files to always be loaded for the customer interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Loader::Customer::CommonCSS::IE8'}->{'000-Framework'} = [];</pre>

B.1.25.35. Loader::Customer::CommonJS###000-Framework

Description	Value
Description:	List of JS files to always be loaded for the customer interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1

Description	Value
Required:	1
Config-Setting:	<pre>\$Self->{'Loader::Customer::CommonJS'}->{'000-Framework'} = ['thirdparty/jquery-1.4.4/jquery.js', 'thirdparty/jquery-validate-1.7/jquery.validate.js', 'thirdparty/jquery-ui-1.8.7/jquery-ui.js', 'thirdparty/stacktrace/stacktrace.js', 'Core.Debug.js', 'Core.Data.js', 'Core.Exception.js', 'Core.App.js', 'Core.JavaScriptEnhancements.js', 'Core.Config.js', 'Core.UI.js', 'Core.UI.IE7Fixes.js', 'Core.UI.Accessibility.js', 'Core.UI.Dialog.js', 'Core.UI.RichTextEditor.js', 'Core.UI.Datepicker.js', 'Core.UI.Popup.js', 'Core.Form.js', 'Core.Form.ErrorTooltips.js', 'Core.Form.Validate.js', 'Core.Customer.js'];</pre>

B.1.25.36. Loader::Agent::DefaultSelectedSkin

Description	Value
Description:	The agent skin's InternalName which should be used in the agent interface. Please check the available skins in Frontend::Agent::Skins.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Loader::Agent::DefaultSelectedSkin'} = 'default';</pre>

B.1.25.37. Loader::Customer::SelectedSkin::HostBased

Description	Value
Description:	It is possible to configure different skins, for example to distinguish between different customers, to be used on a per-domain basis within the application. Using a regular expression (regex), you can configure a Key/Content pair to match a domain. The value in "Key" should match the domain, and the value in "Content" should be a valid skin on your system. Please see the example entries for the proper form of the regex.
Group:	Framework
SubGroup:	Core::Web
Valid:	0

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'Loader::Customer::SelectedSkin::HostBased'} = { 'host1\\.example\\.com' => 'Someskin1', 'host2\\.example\\.com' => 'Someskin2' };</pre>

B.1.26. Core::WebUserAgent

B.1.26.1. WebUserAgent::Timeout

Description	Value
Description:	Sets the timeout (in seconds) for http/ftp downloads.
Group:	Framework
SubGroup:	Core::WebUserAgent
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'WebUserAgent::Timeout'} = '15';</pre>

B.1.26.2. WebUserAgent::Proxy

Description	Value
Description:	Defines the connections for http/ftp, via a proxy.
Group:	Framework
SubGroup:	Core::WebUserAgent
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'WebUserAgent::Proxy'} = 'http://proxy.sn.no:8001/';</pre>

B.1.27. Crypt::PGP

B.1.27.1. PGP

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

Description	Value
Description:	Enables PGP support. When PGP support is enabled for signing and securing mail, it is HIGHLY recommended that the web server be run as the OTRS user. Otherwise, there will be problems with the privileges when accessing .gnupg folder.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PGP'} = '0';</code>

B.1.27.2. PGP::Bin

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

B.1.27.3. PGP::Options

Description	Value
Description:	Sets the options for PGP binary.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PGP::Options'} = '--homedir /opt/otrs/.gnupg/ --batch --no-tt</code>

B.1.27.4. PGP::Key::Password

Description	Value
Description:	Sets the password for private PGP key.
Group:	Framework
SubGroup:	Crypt::PGP

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

B.1.27.5. PGP::TrustedNetwork

Description	Value
Description:	Set this to yes if you trust in all your public and private pgp keys, even if they are not certified with a trusted signature.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PGP::TrustedNetwork'} = '0';</pre>

B.1.27.6. PGP::Log

Description	Value
Description:	Configure your own log text for PGP.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'PGP::Log'} = { 'BADSIG' => 'The PGP signature with the keyid has not been verified suc 'ERRSIG' => 'It was not possible to check the PGP signature, this may b 'EXPKEYSIG' => 'The PGP signature was made by an expired key.', 'GOODSIG' => 'Good PGP signature.', 'KEYREVOKED' => 'The PGP signature was made by a revoked key, this coul 'NODATA' => 'No valid OpenPGP data found.', 'NO_PUBKEY' => 'No public key found.', 'REVKEYSIG' => 'The PGP signature was made by a revoked key, this could 'SIGEXPIRED' => 'The PGP signature is expired.', 'SIG_ID' => 'Signature data.', 'TRUST_UNDEFINED' => 'This key is not certified with a trusted signatur 'VALIDSIG' => 'The PGP signature with the keyid is good.' };</pre>

B.1.28. Crypt::SMIME

B.1.28.1. SMIME

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

B.1.28.2. SMIME::Bin

Description	Value
Description:	Defines the path to open ssl binary. It may need a HOME env (\$ENV{HOME} = '/var/lib/wwwrun';).
Group:	Framework
SubGroup:	Crypt::SMIME
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SMIME::Bin'} = '/usr/bin/openssl';</code>

B.1.28.3. SMIME::CertPath

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

B.1.28.4. SMIME::PrivatePath

Description	Value
Description:	Specifies the directory where private SSL certificates are stored.
Group:	Framework
SubGroup:	Crypt::SMIME

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

B.1.29. Frontend::Admin::ModuleRegistration

B.1.29.1. Frontend::Module###Admin

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'Admin'} = { 'Description' => 'Admin-Area', 'Group' => ['admin'], 'Loader' => { 'CSS' => ['Core.Agent.Admin.css'], 'CSS_IE7' => ['Core.Agent.AdminIE7.css'], 'JavaScript' => ['Core.Agent.Admin.SysConfig.js'] }, 'NavBar' => [{ 'AccessKey' => 'a', 'Block' => 'ItemArea', 'Description' => "", 'Link' => 'Action=Admin', 'LinkOption' => "", 'Name' => 'Admin', 'NavBar' => 'Admin', 'Prio' => '10000', 'Type' => 'Menu' }], 'NavBarModule' => { 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin' }, 'NavBarName' => 'Admin', 'Title' => "" }; </pre>

B.1.29.2. Frontend::Module###AdminInit

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminInit'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarName' => "", 'Title' => 'Init' }; </pre>

B.1.29.3. Frontend::Module###AdminUser

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminUser'} = { 'Description' => 'Create and manage agents.', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Agent', 'Description' => 'Create and manage agents.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Agents', 'Prio' => '100' }, 'NavBarName' => 'Admin', 'Title' => 'Agents' }; </pre>

B.1.29.4. Frontend::Module###AdminGroup

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminGroup'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Agent', 'Description' => 'Create and manage groups.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Groups', 'Prio' => '150' }, 'NavBarName' => 'Admin', 'Title' => 'Groups' }; </pre>

B.1.29.5. Frontend::Module###AdminUserGroup

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminUserGroup'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Agent', 'Description' => 'Link agents to groups.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Agents <-> Groups', 'Prio' => '200' }, 'NavBarName' => 'Admin', 'Title' => 'Agents <-> Groups' };</pre>

B.1.29.6. Frontend::Module###AdminCustomerUser

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminCustomerUser'} = { 'Description' => 'Edit Customers', 'Group' => ['admin', 'users'], 'GroupRo' => ["], 'Loader' => { 'JavaScript' => ['Core.Agent.TicketAction.js'] }, 'NavBar' => [{ 'AccessKey' => 'c', 'Block' => 'ItemArea', 'Description' => "", 'Link' => 'Action=AdminCustomerUser;Nav=Agent', 'LinkOption' => "", 'Name' => 'Customers', 'NavBar' => 'Customers', 'Prio' => '9000', 'Type' => 'Menu' }], 'NavBarModule' => { 'Block' => 'Customer', 'Description' => 'Create and manage customers.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Customers', 'Prio' => '300' }, 'NavBarName' => 'Customers', 'Title' => 'Customers' };</pre>

B.1.29.7. Frontend::Module###AdminCustomerCompany

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminCustomerCompany'} = { 'Description' => 'Admin', 'Group' => ['admin', 'users'], 'GroupRo' => ["], 'NavBar' => [{ 'AccessKey' => 'c', 'Block' => 'ItemArea', 'Description' => "", 'Link' => 'Action=AdminCustomerCompany;Nav=Agent', 'LinkOption' => "", 'Name' => 'Companies', 'NavBar' => 'Companies', 'Prio' => '9100', 'Type' => 'Menu' }], 'NavBarModule' => { 'Block' => 'Customer', 'Description' => 'Create and manage companies.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Customer Companies', 'Prio' => '310' }, 'NavBarName' => 'Admin', 'Title' => 'Customer Companies' }; </pre>

B.1.29.8. Frontend::Module###AdminCustomerUserGroup

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminCustomerUserGroup'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Customer', 'Description' => 'Link customers to groups.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Customers <-> Groups', 'Prio' => '400' }, 'NavBarName' => 'Admin', 'Title' => 'Customers <-> Groups' }; </pre>

B.1.29.9. Frontend::Module###AdminCustomerUserService

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminCustomerUserService'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Customer', 'Description' => 'Link customers to services.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Customers <-> Services', 'Prio' => '500' }, 'NavBarName' => 'Admin', 'Title' => 'Customers <-> Services' }; </pre>

B.1.29.10. Frontend::Module###AdminRole

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminRole'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Agent', 'Description' => 'Create and manage roles.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Roles', 'Prio' => '600' }, 'NavBarName' => 'Admin', 'Title' => 'Role' }; </pre>

B.1.29.11. Frontend::Module###AdminRoleUser

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminRoleUser'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Agent', 'Description' => 'Link agents to roles.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Agents <-> Roles', 'Prio' => '700' }, 'NavBarName' => 'Admin', 'Title' => 'Roles <-> Agents' };</pre>

B.1.29.12. Frontend::Module###AdminRoleGroup

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminRoleGroup'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Agent', 'Description' => 'Link roles to groups.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Roles <-> Groups', 'Prio' => '800' }, 'NavBarName' => 'Admin', 'Title' => 'Roles <-> Groups' };</pre>

B.1.29.13. Frontend::Module###AdminSMIME

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminSMIME'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Email', 'Description' => 'Manage S/MIME certificates for email encryption.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'S/MIME Certificates', 'Prio' => '1100' }, 'NavBarName' => 'Admin', 'Title' => 'S/MIME Management' }; </pre>

B.1.29.14. Frontend::Module###AdminPGP

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminPGP'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Email', 'Description' => 'Manage PGP keys for email encryption.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'PGP Keys', 'Prio' => '1200' }, 'NavBarName' => 'Admin', 'Title' => 'PGP Key Management' }; </pre>

B.1.29.15. Frontend::Module###AdminMailAccount

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminMailAccount'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Email', 'Description' => 'Manage POP3 or IMAP accounts to fetch email from.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'PostMaster Mail Accounts', 'Prio' => '100' }, 'NavBarName' => 'Admin', 'Title' => 'Mail Account' }; </pre>

B.1.29.16. Frontend::Module###AdminPostMasterFilter

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

B.1.29.17. Frontend::Module###AdminEmail

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

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

B.1.29.18. Frontend::Module###AdminSession

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

B.1.29.19. Frontend::Module###AdminPerformanceLog

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminPerformanceLog'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'Loader' => { 'CSS' => ['Core.Agent.Admin.PerformanceLog.css'] }, 'NavBarModule' => { 'Block' => 'System', 'Description' => 'View performance benchmark results.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Performance Log', 'Prio' => '550' }, 'NavBarName' => 'Admin', 'Title' => 'Performance Log' }; </pre>

B.1.29.20. Frontend::Module###AdminLog

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminLog'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'System', 'Description' => 'View system log messages.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'System Log', 'Prio' => '600' }, 'NavBarName' => 'Admin', 'Title' => 'System Log' }; </pre>

B.1.29.21. Frontend::Module###AdminSelectBox

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminSelectBox'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'System', 'Description' => 'Execute SQL statements.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'SQL Box', 'Prio' => '700' }, 'NavBarName' => 'Admin', 'Title' => 'SQL Box' }; </pre>

B.1.29.22. Frontend::Module###AdminPackageManager

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminPackageManager'} = { 'Description' => 'Software Package Manager', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'System', 'Description' => 'Update and extend your system with software packages.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Package Manager', 'Prio' => '1000' }, 'NavBarName' => 'Admin', 'Title' => 'Package Manager' }; </pre>

B.1.30. Frontend::Agent

B.1.30.1. AgentLogo

Description	Value
Description:	The logo shown in the header of the agent interface. The URL to the image must be a relative URL to the skin image directory.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'AgentLogo'} = { 'StyleHeight' => '67px', 'StyleRight' => '38px', 'StyleTop' => '-4px', 'StyleWidth' => '244px', 'URL' => 'skins/Agent/default/img/logo_bg.png' }; </pre>

B.1.30.2. AgentLoginLogo

Description	Value
Description:	The logo shown on top of the login box of the agent interface. The URL to the image must be relative URL to the skin image directory.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'AgentLoginLogo'} = { 'StyleHeight' => '100px', 'URL' => 'skins/Agent/default/img/loginlogo_default.png' }; </pre>

B.1.30.3. LoginURL

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

B.1.30.4. LogoutURL

Description	Value
Description:	Defines an alternate URL, where the logout link refers to.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0

Description	Value
Required:	0
Config-Setting:	<code>\$Self->{'LogoutURL'} = 'http://host.example.com/thanks-for-using-otrs.'</code>

B.1.30.5. PreApplicationModule###AgentInfo

Description	Value
Description:	Defines a useful module to load specific user options or to display news.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'PreApplicationModule'}->{'AgentInfo'} = 'Kernel::Modules::AgentInfo';</code>

B.1.30.6. InfoKey

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

B.1.30.7. InfoFile

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

B.1.30.8. LostPassword

Description	Value
Description:	Activates lost password feature for agents, in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LostPassword'} = '1';</code>

B.1.30.9. ShowMotd

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

B.1.30.10. NotificationSubjectLostPasswordToken

Description	Value
Description:	Defines the subject for notification mails sent to agents, with token about new requested password.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'NotificationSubjectLostPasswordToken'} = 'New OTRS password r</code>

B.1.30.11. NotificationBodyLostPasswordToken

Description	Value
Description:	Defines the body text for notification mails sent to agents, with token about new requested password (after using this link the new password will be sent).
Group:	Framework

Description	Value
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'NotificationBodyLostPasswordToken'} = 'Hi <OTRS_USERFIRSTNAME> You or someone impersonating you has requested to change your OTRS password. If you want to do this, click on the link below. You will receive another <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/<OTRS_CONFIG_ScriptAlias>in If you did not request a new password, please ignore this email. ';</pre>

B.1.30.12. NotificationSubjectLostPassword

Description	Value
Description:	Defines the subject for notification mails sent to agents, about new password.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'NotificationSubjectLostPassword'} = 'New OTRS password';</pre>

B.1.30.13. NotificationBodyLostPassword

Description	Value
Description:	Defines the body text for notification mails sent to agents, about new password (after using this link the new password will be sent).
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'NotificationBodyLostPassword'} = 'Hi <OTRS_USERFIRSTNAME>, Here\'s your new OTRS password. New password: <OTRS_NEWPW> You can log in via the following URL: <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/<OTRS_CONFIG_ScriptAlias>in ';</pre>

B.1.30.14. NewTicketInNewWindow::Enabled

Description	Value
Description:	If enabled, TicketPhone and TicketEmail will be open in new windows.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'NewTicketInNewWindow::Enabled'} = '0';</code>

B.1.30.15. OpenMainMenuOnHover

Description	Value
Description:	If enabled, the first level of the main menu opens on mouse hover (instead of click only).
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'OpenMainMenuOnHover'} = '0';</code>

B.1.30.16. Loader::Agent::Skin###000-default

Description	Value
Description:	Default skin for interface.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Loader::Agent::Skin'}->{'000-default'} = { 'Description' => 'This is the default orange - black skin.', 'HomePage' => 'www.otrs.org', 'InternalName' => 'default', 'VisibleName' => 'Default' };</code>

B.1.30.17. Loader::Agent::Skin###001-ivory

Description	Value
Description:	Balanced white skin by Felix Niklas.

Description	Value
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Loader::Agent::Skin'}->{'001-ivory'} = { 'Description' => 'Balanced white skin by Felix Niklas', 'HomePage' => 'www.felixniklas.de', 'InternalName' => 'ivory', 'VisibleName' => 'Ivory' };</pre>

B.1.30.18. Loader::Agent::Skin###001-slim

Description	Value
Description:	Experimental "Slim" skin which tries to save screen space for power users.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Loader::Agent::Skin'}->{'001-slim'} = { 'Description' => 'Experimental "Slim" skin which tries to save screen s 'HomePage' => 'www.otrs.org', 'InternalName' => 'slim', 'VisibleName' => 'Slim' };</pre>

B.1.30.19. Loader::Agent::DefaultSelectedSkin::HostBased

Description	Value
Description:	It is possible to configure different skins, for example to distinguish between diferent agents, to be used on a per-domain basis within the application. Using a regular expression (regex), you can configure a Key/Content pair to match a domain. The value in "Key" should match the domain, and the value in "Content" should be a valid skin on your system. Please see the example entries for the proper form of the regex.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Loader::Agent::DefaultSelectedSkin::HostBased'} = { 'host1\\.example\\.com' => 'SomeSkin1', 'host2\\.example\\.com' => 'SomeSkin2' };</pre>

B.1.31. Frontend::Agent::Dashboard

B.1.31.1. DashboardBackend###0000-ProductNotify

Description	Value
Description:	Defines the parameters for the dashboard backend. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" defines the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'DashboardBackend'}->{'0000-ProductNotify'} = { 'Block' => 'ContentLarge', 'CacheTTLLocal' => '1440', 'Default' => '1', 'Description' => 'News about OTRS releases!', 'Group' => 'admin', 'Module' => 'Kernel::Output::HTML::DashboardProductNotify', 'Title' => 'Product News', 'URL' => 'http://otrs.org/product.xml' };</pre>

B.1.31.2. DashboardBackend###0400-UserOnline

Description	Value
Description:	Defines the parameters for the dashboard backend. "Limit" defines the number of entries displayed by default. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" defines the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'DashboardBackend'}->{'0400-UserOnline'} = { 'Block' => 'ContentSmall', 'CacheTTLocal' => '5', 'Default' => '0', 'Description' => "", 'Filter' => 'Agent', 'Group' => "", 'IdleMinutes' => '60', 'Limit' => '10', 'Module' => 'Kernel::Output::HTML::DashboardUserOnline', 'ShowEmail' => '1', 'SortBy' => 'UserLastname', 'Title' => 'Online' }; </pre>

B.1.31.3. DashboardBackend###0410-RSS

Description	Value
Description:	Defines the parameters for the dashboard backend. "Limit" defines the number of entries displayed by default. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTL" indicates the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'DashboardBackend'}->{'0410-RSS'} = { 'Block' => 'ContentSmall', 'CacheTTL' => '360', 'Default' => '1', 'Description' => "", 'Group' => "", 'Limit' => '6', 'Module' => 'Kernel::Output::HTML::DashboardRSS', 'Title' => 'OTRS News', 'URL' => 'http://otrs.org/rss/' }; </pre>

B.1.31.4. DashboardBackend###0200-Image

Description	Value
Description:	Defines the parameters for the dashboard backend. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTL" indicates the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'DashboardBackend'}->{'0200-Image'} = { 'Block' => 'ContentLarge', 'Default' => '1', 'Description' => 'Some picture description!', 'Group' => "", 'Height' => '140', 'Link' => 'http://otrs.org/', 'LinkTitle' => 'http://otrs.org/', 'Module' => 'Kernel::Output::HTML::DashboardImage', 'Title' => 'A picture', 'URL' => 'http://www.otrs.com/uploads/pics/jointhecommunity_02.jpg', 'Width' => '198' }; </pre>

B.1.31.5. DashboardBackend###0210-MOTD

Description	Value
Description:	Shows the message of the day (MOTD) in the agent dashboard. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'DashboardBackend'}->{'0210-MOTD'} = { 'Block' => 'ContentLarge', 'Default' => '1', 'Group' => "", 'Module' => 'Kernel::Output::HTML::DashboardMOTD', 'Title' => 'Message of the Day' }; </pre>

B.1.31.6. DashboardBackend###0300-IFrame

Description	Value
Description:	Defines the parameters for the dashboard backend. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTL" indicates the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'DashboardBackend'}->{'0300-IFrame'} = { 'Align' => 'left', 'Block' => 'ContentLarge', 'Default' => '1', 'Description' => 'Some description!', 'Frameborder' => '1', 'Group' => "", 'Height' => '800', 'Link' => 'http://otrs.org/', 'LinkTitle' => 'OTRS.org/', 'Marginheight' => '5', 'Marginwidth' => '5', 'Module' => 'Kernel::Output::HTML::DashboardIFrame', 'Scrolling' => 'auto', 'Title' => 'A Website', 'URL' => 'http://www.otrs.org/', 'Width' => '1024' }; </pre>

B.1.32. Frontend::Agent::ModuleMetaHead

B.1.32.1. Frontend::HeaderMetaModule###1-Refresh

Description	Value
Description:	Defines the module to generate html refresh headers of html sites.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleMetaHead
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Frontend::HeaderMetaModule'}->{'1-Refresh'} = { 'Module' => 'Kernel::Output::HTML::HeaderMetaRefresh' }; </pre>

B.1.33. Frontend::Agent::ModuleNotify

B.1.33.1. 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

Description	Value
Config-Setting:	<pre>\$Self->{'Frontend::NotifyModule'}->{'1-CharsetCheck'} = { 'Module' => 'Kernel::Output::HTML::NotificationCharsetCheck' };</pre>

B.1.33.2. Frontend::NotifyModule###2-UID-Check

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

B.1.33.3. Frontend::NotifyModule###3-ShowAgentOnline

Description	Value
Description:	Defines the module that shows all the currently logged in agents in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::NotifyModule'}->{'3-ShowAgentOnline'} = { 'IdleMinutes' => '60', 'Module' => 'Kernel::Output::HTML::NotificationAgentOnline', 'ShowEmail' => '1' };</pre>

B.1.33.4. Frontend::NotifyModule###4-ShowCustomerOnline

Description	Value
Description:	Defines the module that shows all the currently logged in customers in the agent interface.
Group:	Framework

Description	Value
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::NotifyModule'}->{'4-ShowCustomerOnline'} = { 'IdleMinutes' => '60', 'Module' => 'Kernel::Output::HTML::NotificationCustomerOnline', 'ShowEmail' => '1' }; </pre>

B.1.34. Frontend::Agent::ModuleRegistration

B.1.34.1. Frontend::Module###Logout

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

B.1.34.2. Frontend::Module###AgentDashboard

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

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentDashboard'} = { 'Description' => 'Agent Dashboard', 'Loader' => { 'JavaScript' => ['thirdparty/flot-0.6/excanvas.js', 'thirdparty/flot-0.6/jquery.flot.js', 'Core.UI.Chart.js', 'Core.UI.DnD.js', 'Core.Agent.Dashboard.js'] }, 'NavBar' => [{ 'AccessKey' => 'd', 'Block' => 'ItemArea', 'Description' => "", 'Link' => 'Action=AgentDashboard', 'LinkOption' => "", 'Name' => 'Dashboard', 'NavBar' => 'Dashboard', 'Prio' => '50', 'Type' => 'Menu' }], 'NavBarName' => 'Dashboard', 'Title' => "" }; </pre>

B.1.34.3. Frontend::Module###AgentPreferences

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

B.1.34.4. Frontend::Module###PictureUpload

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'PictureUpload'} = { 'Description' => 'Picture upload module', 'NavBarName' => 'Ticket', 'Title' => 'Picture-Upload' }; </pre>

B.1.34.5. Frontend::Module###AgentSpelling

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

B.1.34.6. Frontend::Module###SpellingInline

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

B.1.34.7. Frontend::Module###AgentBook

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

B.1.34.8. Frontend::Module###AgentLinkObject

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

B.1.34.9. Frontend::Module###AgentInfo

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

B.1.34.10. Frontend::Module###AgentSearch

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

B.1.34.11. CustomerFrontend::Module###SpellingInline

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

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

B.1.34.12. Frontend::Module###AgentHTMLReference

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentHTMLReference'} = { 'Description' => 'HTML Reference', 'Group' => ['users'], 'GroupRo' => ['users'], 'Loader' => { 'CSS' => ['Core.Agent.HTMLReference.css'] }, 'NavBarName' => "", 'Title' => 'HTML Reference' }; </pre>

B.1.34.13. Frontend::Module###AgentStats

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

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentStats'} = { 'Description' => 'Stats', 'Group' => ['stats'], 'GroupRo' => ['stats'], 'Loader' => { 'JavaScript' => ['Core.Agent.Stats.js'] }, 'NavBar' => [{ 'AccessKey' => "", 'Block' => 'ItemArea', 'Description' => "", 'Link' => 'Action=AgentStats;Subaction=Overview', 'LinkOption' => "", 'Name' => 'Statistics', 'NavBar' => 'Stats', 'Prio' => '8500', 'Type' => 'Menu' }, { 'AccessKey' => "", 'Block' => "", 'Description' => 'Overview', 'GroupRo' => ['stats'], 'Link' => 'Action=AgentStats;Subaction=Overview', 'LinkOption' => "", 'Name' => 'Overview', 'NavBar' => 'Stats', 'Prio' => '100', 'Type' => "" }, { 'AccessKey' => "", 'Block' => "", 'Description' => 'New', 'Group' => ['stats'], 'Link' => 'Action=AgentStats;Subaction=Add', 'LinkOption' => "", 'Name' => 'New', 'NavBar' => 'Stats', 'Prio' => '200', 'Type' => "" }, { 'AccessKey' => "", 'Block' => "", 'Description' => 'Import', 'Group' => ['stats'], 'Link' => 'Action=AgentStats;Subaction=Import', 'LinkOption' => "", 'Name' => 'Import', 'NavBar' => 'Stats', 'Prio' => '300', 'Type' => "" }], 'NavBarName' => 'Stats', 'Title' => 'Stats' }; </pre>

B.1.35. Frontend::Agent::NavBarModule

B.1.35.1. Frontend::NavBarModule###6-CustomerCompany

Description	Value
Description:	Frontend module registration (disable company link if no company feature is used).
Group:	Framework
SubGroup:	Frontend::Agent::NavBarModule
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Frontend::NavBarModule'}->{'6-CustomerCompany'} = { 'Module' => 'Kernel::Output::HTML::NavBarCustomerCompany' };</pre>

B.1.36. Frontend::Agent::Preferences

B.1.36.1. PreferencesTableValue

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

B.1.36.2. PreferencesTableUserID

Description	Value
Description:	Defines the name of the column to store the user identifier in the preferences table.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'PreferencesTableUserID'} = 'user_id';</pre>

B.1.36.3. PreferencesView

Description	Value
Description:	Sets the display order of the different items in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'PreferencesView'} = ['User Profile', 'Email Settings', 'Other Settings'];</code>

B.1.36.4. PreferencesGroups###Password

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'PreferencesGroups'}->{'Password'} = { 'Active' => '1', 'Area' => 'Agent', 'Column' => 'User Profile', 'Label' => 'Change password', 'Module' => 'Kernel::Output::HTML::PreferencesPassword', 'PasswordMaxLoginFailed' => '0', 'PasswordMin2Characters' => '0', 'PasswordMin2Lower2UpperCharacters' => '0', 'PasswordMinSize' => '0', 'PasswordNeedDigit' => '0', 'PasswordRegExp' => "", 'Prio' => '0500' };</code>

B.1.36.5. PreferencesGroups###SpellDict

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view. Take care to maintain the dictionaries installed in the system in the data section.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'SpellDict'} = { 'Active' => '1', 'Column' => 'User Profile', 'Data' => { 'deutsch' => 'Deutsch', 'english' => 'English' }, 'DataSelected' => 'english', 'Key' => 'Default spelling dictionary', 'Label' => 'Spelling Dictionary', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserSpellDict', 'Prio' => '2000' }; </pre>

B.1.36.6. PreferencesGroups###Comment

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'Comment'} = { 'Active' => '0', 'Block' => 'Input', 'Column' => 'Other Settings', 'Data' => '\$Env{"UserComment"}', 'Key' => 'Comment', 'Label' => 'Comment', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserComment', 'Prio' => '6000' }; </pre>

B.1.36.7. PreferencesGroups###FreeText

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'FreeText'} = { 'Active' => '1', 'Block' => 'Input', 'Column' => 'Other Settings', 'Data' => '\$Env{"UserFreeText"}', 'Key' => 'Example for free text', 'Label' => 'Example for free text', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserFreeText', 'Prio' => '7000' }; </pre>

B.1.36.8. PreferencesGroups###Language

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'Language'} = { 'Active' => '1', 'Column' => 'User Profile', 'Key' => 'Frontend language', 'Label' => 'Language', 'Module' => 'Kernel::Output::HTML::PreferencesLanguage', 'PrefKey' => 'UserLanguage', 'Prio' => '1000' }; </pre>

B.1.36.9. PreferencesGroups###Skin

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'Skin'} = { 'Active' => '1', 'Column' => 'User Profile', 'Key' => 'Wear this frontend skin', 'Label' => 'Skin', 'Module' => 'Kernel::Output::HTML::PreferencesSkin', 'PrefKey' => 'UserSkin', 'Prio' => '2000' }; </pre>

B.1.36.10. PreferencesGroups###Theme

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'Theme'} = { 'Active' => '1', 'Column' => 'User Profile', 'Key' => 'Frontend theme', 'Label' => 'Theme', 'Module' => 'Kernel::Output::HTML::PreferencesTheme', 'PrefKey' => 'UserTheme', 'Prio' => '3000' };</pre>

B.1.36.11. PreferencesGroups###OutOfOffice

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'OutOfOffice'} = { 'Active' => '1', 'Block' => 'OutOfOffice', 'Column' => 'User Profile', 'Key' => "", 'Label' => 'Out Of Office Time', 'Module' => 'Kernel::Output::HTML::PreferencesOutOfOffice', 'PrefKey' => 'UserOutOfOffice', 'Prio' => '4000' };</pre>

B.1.36.12. PreferencesGroups###TimeZone

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'TimeZone'} = { 'Active' => '1', 'Column' => 'User Profile', 'Key' => 'Time Zone', 'Label' => 'Time Zone', 'Module' => 'Kernel::Output::HTML::PreferencesTimeZone', 'PrefKey' => 'UserTimeZone', 'Prio' => '5000' }; </pre>

B.1.36.13. PreferencesGroups###CSVSeparator

Description	Value
Description:	Gives end users the possibility to override the separator character for CSV files, defined in the translation files.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'CSVSeparator'} = { 'Active' => '1', 'Column' => 'Other Settings', 'Data' => { " => ",", ',' => ',', ';' => ';', '\\t' => 'tab', ' ' => ' ' }, 'DataSelected' => '0', 'Desc' => 'Select the separator character used in CSV files (stats and 'Key' => 'CSV Separator', 'Label' => 'CSV Separator', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserCSVSeparator', 'Prio' => '4000' }; </pre>

B.1.37. Frontend::Agent::SearchRouter

B.1.37.1. Frontend::SearchDefault

Description	Value
Description:	Search backend default router.
Group:	Framework
SubGroup:	Frontend::Agent::SearchRouter
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::SearchDefault'} = 'Action=AgentTicketSearch;Subacti </pre>

B.1.38. Frontend::Agent::Stats

B.1.38.1. Stats::SearchPageShown

Description	Value
Description:	Defines the default maximum number of search results shown on the overview page.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::SearchPageShown'} = '20';</code>

B.1.38.2. Stats::DefaultSelectedDynamicObject

Description	Value
Description:	Defines the default selection at the drop down menu for dynamic objects (Form: Common Specification).
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::DefaultSelectedDynamicObject'} = 'Ticket';</code>

B.1.38.3. Stats::DefaultSelectedPermissions

Description	Value
Description:	Defines the default selection at the drop down menu for permissions (Form: Common Specification).
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::DefaultSelectedPermissions'} = ['stats'];</code>

B.1.38.4. Stats::DefaultSelectedFormat

Description	Value
Description:	Defines the default selection at the drop down menu for stats format (Form: Common Specification). Please insert the format key (see Stats::Format).
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::DefaultSelectedFormat'} = ['Print', 'CSV'];</code>

B.1.38.5. Stats::SearchLimit

Description	Value
Description:	Defines the search limit for the stats.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::SearchLimit'} = '500';</code>

B.1.38.6. Stats::Format

Description	Value
Description:	Defines all the possible stats output formats.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Format'} = { 'CSV' => 'CSV', 'GD::Graph::area' => 'graph-area', 'GD::Graph::bars' => 'graph-bars', 'GD::Graph::hbars' => 'graph-hbars', 'GD::Graph::lines' => 'graph-lines', 'GD::Graph::linespoints' => 'graph-lines-points', 'GD::Graph::pie' => 'graph-pie', 'GD::Graph::points' => 'graph-points', 'Print' => 'Print' };</code>

B.1.38.7. Stats::GraphSize

Description	Value
Description:	Sets the size of the statistic graph.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Stats::GraphSize'} = { '1200x800' => '1200x800', '1600x1200' => '1600x1200', '800x600' => '800x600' };</pre>

B.1.38.8. Stats::TimeType

Description	Value
Description:	Sets the time type which should be shown.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Stats::TimeType'} = 'Extended';</pre>

B.1.38.9. Stats::ExchangeAxis

Description	Value
Description:	Allows agents to exchange the axis of a stat if they generate one.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Stats::ExchangeAxis'} = '0';</pre>

B.1.38.10. Stats::UseAgentElementInStats

Description	Value
Description:	Allows agents to generate individual-related stats.
Group:	Framework
SubGroup:	Frontend::Agent::Stats

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

B.1.38.11. Stats::CustomerIDAsMultiSelect

Description	Value
Description:	Shows all the customer identifiers in a multi-select field (not useful if you have a lot of customer identifiers).
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Stats::CustomerIDAsMultiSelect'} = '1';</code>

B.1.39. Frontend::Customer

B.1.39.1. CustomerHeadline

Description	Value
Description:	The headline shown in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerHeadline'} = 'Example Company Support';</code>

B.1.39.2. CustomerLogo

Description	Value
Description:	The logo shown in the header of the customer interface. The URL to the image must be a relative URL to the skin image directory.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerLogo'} = { 'StyleHeight' => '50px', 'StyleRight' => '25px', 'StyleTop' => '2px', 'StyleWidth' => '135px', 'URL' => 'skins/Customer/default/img/logo.png' };</pre>

B.1.39.3. CustomerPanelUserID

Description	Value
Description:	Defines the user identifier for the customer panel.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerPanelUserID'} = '1';</pre>

B.1.39.4. CustomerGroupSupport

Description	Value
Description:	Activates support for customer groups.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerGroupSupport'} = '0';</pre>

B.1.39.5. CustomerGroupAlwaysGroups

Description	Value
Description:	Defines the groups every customer user will be in (if CustomerGroupSupport is enabled and you don't want to manage every user for these groups).
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerGroupAlwaysGroups'} = ['users'];</pre>

B.1.39.6. CustomerPanelLoginURL

Description	Value
Description:	Defines an alternate login URL for the customer panel..
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'CustomerPanelLoginURL'} = 'http://host.example.com/cgi-bin/lo</code>

B.1.39.7. CustomerPanelLogoutURL

Description	Value
Description:	Defines an alternate logout URL for the customer panel.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'CustomerPanelLogoutURL'} = 'http://host.example.com/cgi-bin/1</code>

B.1.39.8. Frontend::CustomerUser::Item###1-GoogleMaps

Description	Value
Description:	Defines a customer item, which generates a google maps icon at the end of a customer info block.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Frontend::CustomerUser::Item'}->{'1-GoogleMaps'} = { 'Attributes' => 'UserStreet;UserCity;UserCountry;', 'CSS' => 'Core.Agent.CustomerUser.GoogleMaps.css', 'CSSClass' => 'GoogleMaps', 'Module' => 'Kernel::Output::HTML::CustomerUserGeneric', 'Required' => 'UserStreet;UserCity;', 'Target' => '_blank', 'Text' => 'Location', 'URL' => 'http://maps.google.com/maps?z=7&q=' }; </pre>

B.1.39.9. Frontend::CustomerUser::Item###2-Google

Description	Value
Description:	Defines a customer item, which generates a google icon at the end of a customer info block.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::CustomerUser::Item'}->{'2-Google'} = { 'Attributes' => 'UserFirstname;UserLastname;', 'CSS' => 'Core.Agent.CustomerUser.Google.css', 'CSSClass' => 'Google', 'Module' => 'Kernel::Output::HTML::CustomerUserGeneric', 'Required' => 'UserFirstname;UserLastname;', 'Target' => '_blank', 'Text' => 'Google', 'URL' => 'http://google.com/search?q=' }; </pre>

B.1.39.10. Frontend::CustomerUser::Item###2-LinkedIn

Description	Value
Description:	Defines a customer item, which generates a LinkedIn icon at the end of a customer info block.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::CustomerUser::Item'}->{'2-LinkedIn'} = { 'Attributes' => 'UserFirstname;UserLastname;', 'CSS' => 'Core.Agent.CustomerUser.Linkedin.css', 'CSSClass' => 'LinkedIn', 'Module' => 'Kernel::Output::HTML::CustomerUserGeneric', 'Required' => 'UserFirstname;UserLastname;', 'Target' => '_blank', 'Text' => 'LinkedIn', 'URL' => 'http://www.linkedin.com/commonSearch?type=people&keywords=' }; </pre>

B.1.39.11. Frontend::CustomerUser::Item###3-XING

Description	Value
Description:	Defines a customer item, which generates a XING icon at the end of a customer info block.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::CustomerUser::Item'}->{'3-XING'} = { 'Attributes' => 'UserFirstname;UserLastname;', 'CSS' => 'Core.Agent.CustomerUser.Xing.css', 'CSSClass' => 'Xing', 'Module' => 'Kernel::Output::HTML::CustomerUserGeneric', 'Required' => 'UserFirstname;UserLastname;', 'Target' => '_blank', 'Text' => 'XING', 'URL' => 'https://www.xing.com/app/search?op=search;keywords=' }; </pre>

B.1.39.12. CustomerPanelPreApplicationModule###CustomerAccept

Description	Value
Description:	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

Description	Value
Config-Setting:	<code>\$Self->{'CustomerPanelPreApplicationModule'}->{'CustomerAccept'} = 'Ke</code>

B.1.39.13. CustomerPanel::InfoKey

Description	Value
Description:	Defines the key to check with CustomerAccept. If this user preferences key is true, then the message is accepted by the system.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'CustomerPanel::InfoKey'} = 'CustomerAccept1';</code>

B.1.39.14. CustomerPanel::InfoFile

Description	Value
Description:	Defines the path of the shown info file, that is located under Kernel/Output/HTML/Standard/CustomerAccept.dtl.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'CustomerPanel::InfoFile'} = 'CustomerAccept';</code>

B.1.39.15. CustomerPanelLostPassword

Description	Value
Description:	Activates lost password feature for customers.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelLostPassword'} = '1';</code>

B.1.39.16. CustomerPanelCreateAccount

Description	Value
Description:	Enables customers to create their own accounts.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelCreateAccount'} = '1';</code>

B.1.39.17. CustomerPanelSubjectLostPasswordToken

Description	Value
Description:	Defines the subject for notification mails sent to customers, with token about new requested password.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelSubjectLostPasswordToken'} = 'New OTRS password';</code>

B.1.39.18. CustomerPanelBodyLostPasswordToken

Description	Value
Description:	Defines the body text for notification mails sent to customers, with token about new requested password (after using this link the new password will be sent).
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelBodyLostPasswordToken'} = 'Hi <OTRS_USERFIRSTNAME>! You or someone impersonating you has requested to change your OTRS password. If you want to do this, click on this link. You will receive another email with a new password. <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/<OTRS_CONFIG_ScriptAlias>customer/lostpassword/<OTRS_TOKEN> If you did not request a new password, please ignore this email. ';</code>

B.1.39.19. CustomerPanelSubjectLostPassword

Description	Value
Description:	Defines the subject for notification mails sent to customers, about new password.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelSubjectLostPassword'} = 'New OTRS password';</code>

B.1.39.20. CustomerPanelBodyLostPassword

Description	Value
Description:	Defines the body text for notification mails sent to customers, about new password (after using this link the new password will be sent).
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelBodyLostPassword'} = 'Hi <OTRS_USERFIRSTNAME>, New password: <OTRS_NEWPW> <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/<OTRS_CONFIG_ScriptAlias>cu ';</code>

B.1.39.21. CustomerPanelSubjectNewAccount

Description	Value
Description:	Defines the subject for notification mails sent to customers, about new account.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelSubjectNewAccount'} = 'New OTRS Account!';</code>

B.1.39.22. CustomerPanelBodyNewAccount

Description	Value
Description:	Defines the body text for notification mails sent to customers, about new account.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'CustomerPanelBodyNewAccount'} = 'Hi <OTRS_USERFIRSTNAME>, You or someone impersonating you has created a new OTRS account for you. Full name: <OTRS_USERFIRSTNAME> <OTRS_USERLASTNAME> User name: <OTRS_USERLOGIN> Password : <OTRS_USERPASSWORD> You can log in via the following URL. We encourage you to change your p via the Preferences button after logging in. <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/<OTRS_CONFIG_ScriptAlias>cu '; </pre>

B.1.39.23. Loader::Customer::Skin###000-default

Description	Value
Description:	Default skin for OTRS 3.0 interface.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Loader::Customer::Skin'}->{'000-default'} = { 'Description' => 'This is the default orange - black skin for OTRS 3.0. 'HomePage' => 'www.otrs.org', 'InternalName' => 'default', 'VisibleName' => 'Default' }; </pre>

B.1.39.24. Loader::Customer::SelectedSkin

Description	Value
Description:	The customer skin's InternalName which should be used in the customer interface. Please check the available skins in Frontend::Customer::Skins.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'Loader::Customer::SelectedSkin'} = 'default';</code>

B.1.40. Frontend::Customer::Auth

B.1.40.1. Customer::AuthModule

Description	Value
Description:	Defines the module to authenticate customers.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::DB';</code>

B.1.40.2. Customer::AuthModule::DB::CryptType

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, the crypt type of passwords must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::CryptType'} = 'md5';</code>

B.1.40.3. Customer::AuthModule::DB::Table

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, the name of the table where your customer data should be stored must be specified.
Group:	Framework

Description	Value
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::Table'} = 'customer_user';</code>

B.1.40.4. Customer::AuthModule::DB::CustomerKey

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, the name of the column for the CustomerKey in the customer table must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::CustomerKey'} = 'login';</code>

B.1.40.5. Customer::AuthModule::DB::CustomerPassword

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, the column name for the CustomerPassword in the customer table must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::CustomerPassword'} = 'pw';</code>

B.1.40.6. Customer::AuthModule::DB::DSN

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, the DSN for the connection to the customer table must be specified.

Description	Value
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::DSN'} = 'DBI:mysql:database=custome</code>

B.1.40.7. Customer::AuthModule::DB::User

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, a username to connect to the customer table can be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::User'} = 'some_user';</code>

B.1.40.8. Customer::AuthModule::DB::Password

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, a password to connect to the customer table can be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::Password'} = 'some_password';</code>

B.1.40.9. Customer::AuthModule::DB::Type

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

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, a database driver (normally autodetection is used) can be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::Type'} = 'mysql';</code>

B.1.40.10. Customer::AuthModule::HTTPBasicAuth::Replace

Description	Value
Description:	If "HTTPBasicAuth" was selected for Customer::AuthModule, you can specify to strip leading parts of user names (e. g. for domains like example_domain\user to user).
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::HTTPBasicAuth::Replace'} = 'example_dom</code>

B.1.40.11. Customer::AuthModule::HTTPBasicAuth::ReplaceRegExp

Description	Value
Description:	If "HTTPBasicAuth" was selected for Customer::AuthModule, you can specify (by using a RegExp) to strip parts of REMOTE_USER (e. g. for to remove trailing domains). RegExp-Note, \$1 will be the new Login.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::HTTPBasicAuth::ReplaceRegExp'} = '^(.+?</code>

B.1.40.12. Customer::AuthModule::LDAP::Host

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, the LDAP host can be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::Host'} = 'ldap.example.com';</code>

B.1.40.13. Customer::AuthModule::LDAP::BaseDN

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, the BaseDN must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';</code>

B.1.40.14. Customer::AuthModule::LDAP::UID

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, the user identifier must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::UID'} = 'uid';</code>

B.1.40.15. Customer::AuthModule::LDAP::GroupDN

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

Description	Value
Description:	If "LDAP" was selected for Customer::Authmodule, you can check if the user is allowed to authenticate because he is in a posixGroup, e.g. user needs to be in a group xyz to use OTRS. Specify the group, who may access the system.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::GroupDN'} = 'cn=otrsallow,ou=posix';</code>

B.1.40.16. Customer::AuthModule::LDAP::AccessAttr

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, you can specify access attributes here.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::AccessAttr'} = 'memberUid';</code>

B.1.40.17. Customer::AuthModule::LDAP::UserAttr

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, user attributes can be specified. For LDAP posixGroups use UID, for non LDAP posixGroups use full user DN.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::UserAttr'} = 'UID';</code>

B.1.40.18. Customer::AuthModule::LDAP::SearchUserDN

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule and your users have only anonymous access to the LDAP tree, but you want to search through the data, you can do this with a user who has access to the LDAP directory. Specify the username for this special user here.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::SearchUserDN'} = 'cn=binduser,ou=</code>

B.1.40.19. Customer::AuthModule::LDAP::SearchUserPw

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule and your users have only anonymous access to the LDAP tree, but you want to search through the data, you can do this with a user who has access to the LDAP directory. Specify the password for this special user here.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::SearchUserPw'} = 'some_password';</code>

B.1.40.20. 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

Description	Value
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::AlwaysFilter'} = '(!objectclass=</code>

B.1.40.21. Customer::AuthModule::LDAP::UserSuffix

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule and if you want to add a suffix to every customer login name, specify it here, e. g. you just want to write the username user but in your LDAP directory exists user@domain.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::UserSuffix'} = '@domain.com';</code>

B.1.40.22. Customer::AuthModule::LDAP::Params

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule and special parameters 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:	<code>\$Self->{'Customer::AuthModule::LDAP::Params'} = { 'async' => '0', 'port' => '389', 'timeout' => '120', 'version' => '3' };</code>

B.1.40.23. Customer::AuthModule::LDAP::Die

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, you can specify if the applications will stop if e. g. a connection to a server can't be established due to network problems.
Group:	Framework

Description	Value
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::Die'} = '1';</code>

B.1.40.24. Customer::AuthModule::Radius::Host

Description	Value
Description:	If "Radius" was selected for Customer::AuthModule, the radius host must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::Radius::Host'} = 'radiushost';</code>

B.1.40.25. Customer::AuthModule::Radius::Password

Description	Value
Description:	If "Radius" was selected for Customer::AuthModule, the password to authenticate to the radius host must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::Radius::Password'} = 'radiussecret';</code>

B.1.40.26. Customer::AuthModule::Radius::Die

Description	Value
Description:	If "Radius" was selected for Customer::AuthModule, you can specify if the applications will stop if e. g. a connection to a server can't be established due to network problems.
Group:	Framework

Description	Value
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::Radius::Die'} = '1';</code>

B.1.41. Frontend::Customer::ModuleMetaHead

B.1.41.1. CustomerFrontend::HeaderMetaModule###1-Refresh

Description	Value
Description:	Defines the module to generate html refresh headers of html sites, in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleMetaHead
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerFrontend::HeaderMetaModule'}->{'1-Refresh'} = { 'Module' => 'Kernel::Output::HTML::HeaderMetaRefresh'};</code>

B.1.42. Frontend::Customer::ModuleNotify

B.1.42.1. CustomerFrontend::NotifyModule###1-ShowAgentOnline

Description	Value
Description:	Defines the module that shows the currently logged in agents in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'CustomerFrontend::NotifyModule'}->{'1-ShowAgentOnline'} = { 'IdleMinutes' => '60', 'Module' => 'Kernel::Output::HTML::NotificationAgentOnline', 'ShowEmail' => '1'};</code>

B.1.42.2. CustomerFrontend::NotifyModule###1-ShowCustomerOnline

Description	Value
Description:	Defines the module that shows the currently logged in customers in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::NotifyModule'}->{'1-ShowCustomerOnline'} = 'Module' => 'Kernel::Output::HTML::NotificationCustomerOnline', 'ShowEmail' => '1' ;</pre>

B.1.43. Frontend::Customer::ModuleRegistration**B.1.43.1. CustomerFrontend::Module###Logout**

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}->{'Logout'} = { 'Description' => 'Logout of customer panel', 'NavBarName' => "", 'Title' => "" };</pre>

B.1.43.2. CustomerFrontend::Module###CustomerPreferences

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}->{'CustomerPreferences'} = { 'Description' => 'Customer preferences', 'NavBarName' => "", 'Title' => 'Preferences' }; </pre>

B.1.43.3. CustomerFrontend::Module###CustomerAccept

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}->{'CustomerAccept'} = { 'Description' => 'To accept login infos', 'NavBarName' => "", 'Title' => 'Info' }; </pre>

B.1.43.4. CustomerFrontend::Module###PictureUpload

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}->{'PictureUpload'} = { 'Description' => 'Picture upload module', 'NavBarName' => 'Ticket', 'Title' => 'Picture-Upload' }; </pre>

B.1.44. Frontend::Customer::Preferences

B.1.44.1. PreferencesTable

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

Description	Value
Description:	Defines the name of the table, where the customer preferences are stored.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PreferencesTable'} = 'user_preferences';</code>

B.1.44.2. PreferencesTableKey

Description	Value
Description:	Defines the column to store the keys for the preferences table.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PreferencesTableKey'} = 'preferences_key';</code>

B.1.44.3. CustomerPreferences

Description	Value
Description:	Defines the parameters for the customer preferences table.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPreferences'} = { 'Module' => 'Kernel::System::CustomerUser::Preferences::DB', 'Params' => { 'Table' => 'customer_preferences', 'TableKey' => 'preferences_key', 'TableUserID' => 'user_id', 'TableValue' => 'preferences_value' } };</code>

B.1.44.4. CustomerPreferencesView

Description	Value
Description:	Sets the order of the different items in the customer preferences view.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPreferencesView'} = ['User Profile', 'Other Settings'];</code>

B.1.44.5. CustomerPreferencesGroups###Password

Description	Value
Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'CustomerPreferencesGroups'}->{'Password'} = { 'Active' => '1', 'Area' => 'Customer', 'Column' => 'Other Settings', 'Label' => 'Change password', 'Module' => 'Kernel::Output::HTML::PreferencesPassword', 'PasswordMin2Characters' => '0', 'PasswordMin2Lower2UpperCharacters' => '0', 'PasswordMinSize' => '0', 'PasswordNeedDigit' => '0', 'PasswordRegExp' => "", 'Prio' => '1000' };</code>

B.1.44.6. CustomerPreferencesGroups###Language

Description	Value
Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'CustomerPreferencesGroups'}->{'Language'} = { 'Active' => '1', 'Column' => 'User Profile', 'Key' => 'Your language', 'Label' => 'Interface language', 'Module' => 'Kernel::Output::HTML::PreferencesLanguage', 'PrefKey' => 'UserLanguage', 'Prio' => '2000' };</code>

B.1.44.7. CustomerPreferencesGroups###Theme

Description	Value
Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerPreferencesGroups'}->{'Theme'} = { 'Active' => '0', 'Column' => 'User Profile', 'Key' => 'Select your frontend Theme.', 'Label' => 'Theme', 'Module' => 'Kernel::Output::HTML::PreferencesTheme', 'PrefKey' => 'UserTheme', 'Prio' => '1000' };</pre>

B.1.44.8. CustomerPreferencesGroups###TimeZone

Description	Value
Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerPreferencesGroups'}->{'TimeZone'} = { 'Active' => '1', 'Column' => 'User Profile', 'Key' => 'Time Zone', 'Label' => 'Time Zone', 'Module' => 'Kernel::Output::HTML::PreferencesTimeZone', 'PrefKey' => 'UserTimeZone', 'Prio' => '5000' };</pre>

B.1.44.9. CustomerPreferencesGroups###PGP

Description	Value
Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'CustomerPreferencesGroups'}->{'PGP'} = { 'Active' => '1', 'Column' => 'Other Settings', 'Key' => 'PGP Key Upload', 'Label' => 'PGP Key', 'Module' => 'Kernel::Output::HTML::PreferencesPGP', 'PrefKey' => 'UserPGPKey', 'Prio' => '10000' }; </pre>

B.1.44.10. CustomerPreferencesGroups###SMIME

Description	Value
Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerPreferencesGroups'}->{'SMIME'} = { 'Active' => '1', 'Column' => 'Other Settings', 'Key' => 'S/MIME Certificate Upload', 'Label' => 'S/MIME Certificate', 'Module' => 'Kernel::Output::HTML::PreferencesSMIME', 'PrefKey' => 'UserSMIMEKey', 'Prio' => '11000' }; </pre>

B.1.45. Frontend::Public

B.1.45.1. PublicFrontend::CommonParam###Action

Description	Value
Description:	Defines the default value for the action parameter for the public frontend. The action parameter is used in the scripts of the system.
Group:	Framework
SubGroup:	Frontend::Public
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'PublicFrontend::CommonParam'}->{'Action'} = 'PublicDefault'; </pre>

B.1.46. Frontend::Public::ModuleRegistration

B.1.46.1. PublicFrontend::Module###PublicDefault

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

B.1.46.2. PublicFrontend::Module###PublicRepository

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

B.2. Ticket

B.2.1. Core::ArticleFreeText

B.2.1.1. ArticleFreeKey1

Description	Value
Description:	Defines the free key field number 1 for articles to add a new article attribute.
Group:	Ticket

Description	Value
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'ArticleFreeKey1'} = { 'Work1' => 'Work1' };</code>

B.2.1.2. ArticleFreeKey1::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 1 for articles (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'ArticleFreeKey1::DefaultSelection'} = "";</code>

B.2.1.3. ArticleFreeText1

Description	Value
Description:	Defines the free text field number 1 for articles to add a new article attribute.
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'ArticleFreeText1'} = { "" => '-', 'Bugfix' => 'Bugfix', 'Consulting' => 'Consulting', 'Research' => 'Research' };</code>

B.2.1.4. ArticleFreeText1::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 1 for articles (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'ArticleFreeText1::DefaultSelection'} = 'Research';</code>

B.2.1.5. ArticleFreeKey2

Description	Value
Description:	Defines the free key field number 2 for articles to add a new article attribute.
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'ArticleFreeKey2'} = { 'Work2' => 'Work2' };</code>

B.2.1.6. ArticleFreeKey2::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 2 for articles (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'ArticleFreeKey2::DefaultSelection'} = '';</code>

B.2.1.7. ArticleFreeText2

Description	Value
Description:	Defines the free text field number 2 for articles to add a new article attribute.
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'ArticleFreeText2'} = { '' => '-', 'Bugfix' => 'Bugfix', 'Consulting' => 'Consulting', 'Research' => 'Research' };</code>

B.2.1.8. ArticleFreeText2::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 2 for articles (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'ArticleFreeText2::DefaultSelection'} = 'Research';</code>

B.2.1.9. ArticleFreeKey3

Description	Value
Description:	Defines the free key field number 3 for articles to add a new article attribute.
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'ArticleFreeKey3'} = { 'Work3' => 'Work3' };</code>

B.2.1.10. ArticleFreeKey3::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 3 for articles (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'ArticleFreeKey3::DefaultSelection'} = '';</code>

B.2.1.11. ArticleFreeText3

Description	Value
Description:	Defines the free text field number 3 for articles to add a new article attribute.

Description	Value
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'ArticleFreeText3'} = { " => '- ', 'Bugfix' => 'Bugfix', 'Consulting' => 'Consulting', 'Research' => 'Research' }; </pre>

B.2.1.12. ArticleFreeText3::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 3 for articles (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'ArticleFreeText3::DefaultSelection'} = 'Research'; </pre>

B.2.2. Core::FulltextSearch

B.2.2.1. Ticket::SearchIndexModule

Description	Value
Description:	Helps to extend your articles full-text search (From, To, Cc, Subject and Body search). Runtime will do full-text searches on live data (it works fine for up to 50.000 tickets). StaticDB will strip all articles and will build an index after article creation, increasing fulltext searches about 50%. To create an initial index use "bin/otrs.RebuildFulltextIndex.pl".
Group:	Ticket
SubGroup:	Core::FulltextSearch
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::SearchIndexModule'} = 'Kernel::System::Ticket::Article </pre>

B.2.2.2. Ticket::SearchIndex::Attribute

Description	Value
Description:	Configures the full-text index. Execute "bin/otrs.RebuildFulltextIndex.pl" in order to generate a new index.
Group:	Ticket
SubGroup:	Core::FulltextSearch
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::SearchIndex::Attribute'} = { 'WordCountMax' => '1000', 'WordLengthMax' => '30', 'WordLengthMin' => '3' }; </pre>

B.2.2.3. Ticket::EventModulePost###98-ArticleSearchIndex

Description	Value
Description:	Builds an article index right after the article's creation.
Group:	Ticket
SubGroup:	Core::FulltextSearch
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}->{'98-ArticleSearchIndex'} = { 'Event' => '(ArticleCreate ArticleUpdate)', 'Module' => 'Kernel::System::Ticket::Event::ArticleSearchIndex' }; </pre>

B.2.3. Core::LinkObject**B.2.3.1. LinkObject::PossibleLink###0200**

Description	Value
Description:	Links 2 tickets with a "Normal" type link.
Group:	Ticket
SubGroup:	Core::LinkObject
Valid:	1
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'LinkObject::PossibleLink'}->{'0200'} = { 'Object1' => 'Ticket', 'Object2' => 'Ticket', 'Type' => 'Normal' };</code>

B.2.3.2. LinkObject::PossibleLink###0201

Description	Value
Description:	Links 2 tickets with a "ParentChild" type link.
Group:	Ticket
SubGroup:	Core::LinkObject
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'LinkObject::PossibleLink'}->{'0201'} = { 'Object1' => 'Ticket', 'Object2' => 'Ticket', 'Type' => 'ParentChild' };</code>

B.2.4. Core::PostMaster

B.2.4.1. PostmasterMaxEmails

Description	Value
Description:	Maximal auto email responses to own email-address a day (Loop-Protection).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterMaxEmails'} = '40';</code>

B.2.4.2. PostMasterMaxEmailSize

Description	Value
Description:	Maximal size in KBytes for mails that can be fetched via POP3/POP3S/IMAP/IMAPS (KBytes).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'PostMasterMaxEmailSize'} = '16384';</code>

B.2.4.3. PostMasterReconnectMessage

Description	Value
Description:	The "bin/PostMasterMailAccount.pl" will reconnect to POP3/POP3S/IMAP/IMAPS host after the specified count of messages.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostMasterReconnectMessage'} = '20';</code>

B.2.4.4. LoopProtectionModule

Description	Value
Description:	Default loop protection module.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LoopProtectionModule'} = 'Kernel::System::PostMaster::LoopPro</code>

B.2.4.5. LoopProtectionLog

Description	Value
Description:	Path for the log file (it only applies if "FS" was selected for LoopProtectionModule and it is mandatory).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LoopProtectionLog'} = '<OTRS_CONFIG_Home>/var/log/LoopProtect</code>

B.2.4.6. PostmasterAutoHTML2Text

Description	Value
Description:	Converts HTML mails into text messages.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterAutoHTML2Text'} = '1';</code>

B.2.4.7. PostmasterFollowUpSearchInReferences

Description	Value
Description:	Executes follow up checks on In-Reply-To or References headers for mails that don't have a ticket number in the subject.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterFollowUpSearchInReferences'} = '0';</code>

B.2.4.8. PostmasterFollowUpSearchInBody

Description	Value
Description:	Executes follow up mail body checks in mails that don't have a ticket number in the subject.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterFollowUpSearchInBody'} = '0';</code>

B.2.4.9. PostmasterFollowUpSearchInAttachment

Description	Value
Description:	Executes follow up mail attachments checks in mails that don't have a ticket number in the subject.
Group:	Ticket

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

B.2.4.10. PostmasterFollowUpSearchInRaw

Description	Value
Description:	Executes follow up plain/raw mail checks in mails that don't have a ticket number in the subject.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterFollowUpSearchInRaw'} = '0';</code>

B.2.4.11. PostmasterUserID

Description	Value
Description:	Specifies user id of the postmaster data base.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterUserID'} = '1';</code>

B.2.4.12. PostmasterDefaultQueue

Description	Value
Description:	Defines the postmaster default queue.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterDefaultQueue'} = 'Raw';</code>

B.2.4.13. PostmasterDefaultPriority

Description	Value
Description:	Defines the default priority of new tickets.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterDefaultPriority'} = '3 normal';</code>

B.2.4.14. PostmasterDefaultState

Description	Value
Description:	Defines the default state of new tickets.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterDefaultState'} = 'new';</code>

B.2.4.15. PostmasterFollowUpState

Description	Value
Description:	Defines the state of a ticket if it gets a follow-up.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterFollowUpState'} = 'open';</code>

B.2.4.16. PostmasterFollowUpStateClosed

Description	Value
Description:	Defines the state of a ticket if it gets a follow-up and the ticket was already closed.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0

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

B.2.4.17. PostmasterFollowUpOnUnlockAgentNotifyOnlyToOwner

Description	Value
Description:	Sends agent follow-up notification only to the owner, if a ticket is unlocked (the default is to send the notification to all agents).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterFollowUpOnUnlockAgentNotifyOnlyToOwner'} = '0';</code>

B.2.4.18. PostmasterX-Header

Description	Value
Description:	Defines all the X-headers that should be scanned.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre> \$Self->{'PostmasterX-Header'} = ['From', 'To', 'Cc', 'Reply-To', 'ReplyTo', 'Subject', 'Message-ID', 'Message-Id', 'Resent-To', 'Resent-From', 'Precedence', 'Mailing-List', 'List-Id', 'List-Archive', 'Errors-To', 'References', 'In-Reply-To', 'X-Loop', 'X-Spam-Flag', 'X-Spam-Status', 'X-Spam-Level', 'X-No-Loop', 'X-Priority', 'Importance', 'X-Mailer', 'User-Agent', 'Organization', 'X-Original-To', 'Delivered-To', 'Envelope-To', 'Return-Path', 'X-OTRS-Loop', 'X-OTRS-Info', 'X-OTRS-Priority', 'X-OTRS-Queue', 'X-OTRS-Lock', 'X-OTRS-Ignore', 'X-OTRS-State', 'X-OTRS-State-PendingTime', 'X-OTRS-Type', 'X-OTRS-Service', 'X-OTRS-SLA', '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-TicketKey9', 'X-OTRS-TicketKey10', 'X-OTRS-TicketKey11', 'X-OTRS-TicketKey12', 'X-OTRS-TicketKey13', 'X-OTRS-TicketKey14', 'X-OTRS-TicketKey15', 'X-OTRS-TicketKey16', 'X-OTRS-TicketValue1', 'X-OTRS-TicketValue2', 'X-OTRS-TicketValue3', 'X-OTRS-TicketValue4', 'X-OTRS-TicketValue5', 'X-OTRS-TicketValue6', 'X-OTRS-TicketValue7', 'X-OTRS-TicketValue8', 'X-OTRS-TicketValue9', 'X-OTRS-TicketValue10', 'X-OTRS-TicketValue11', 'X-OTRS-TicketValue12', 'X-OTRS-TicketValue13', 'X-OTRS-TicketValue14', 'X-OTRS-TicketValue15', 'X-OTRS-TicketValue16', 'X-OTRS-TicketTime1', 'X-OTRS-TicketTime2', 'X-OTRS-TicketTime3', 'X-OTRS-TicketTime4', 'X-OTRS-TicketTime5', 'X-OTRS-TicketTime6', 'X-OTRS-FollowUp-Priority', 'X-OTRS-FollowUp-Queue', 'X-OTRS-FollowUp-Lock', 'X-OTRS-FollowUp-State', 'X-OTRS-FollowUp-State-PendingTime', 'X-OTRS-FollowUp-Type', 'X-OTRS-FollowUp-Service', 'X-OTRS-FollowUp-SLA', 'X-OTRS-FollowUp-ArticleKey1', 'X-OTRS-FollowUp-ArticleKey2', 'X-OTRS-FollowUp-ArticleKey3', 'X-OTRS-FollowUp-ArticleValue1', 'X-OTRS-FollowUp-ArticleValue2', 'X-OTRS-FollowUp-ArticleValue3', 'X-OTRS-FollowUp-SenderType', 'X-OTRS-FollowUp-ArticleType', 'X-OTRS-FollowUp-TicketKey1', 'X-OTRS-FollowUp-TicketKey2', 'X-OTRS-FollowUp-TicketKey3', </pre>

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

B.2.4.19. 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
Config-Setting:	<pre>\$Self->{'PostMaster::PreFilterModule'}->{'1-Match'} = { 'Match' => { 'From' => 'noreply@' }, 'Module' => 'Kernel::System::PostMaster::Filter::Match', 'Set' => { 'X-OTRS-Ignore' => 'yes' }, 'StopAfterMatch' => '0' }; </pre>

B.2.4.20. PostMaster::PreFilterModule###2-Match

Description	Value
Description:	Module to filter and manipulate incoming messages. Get a 4 digit number to ticket free text, use regex in Match e. g. From => '(.+?)@.+?', and use () as [***] in Set =>.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'PostMaster::PreFilterModule'}->{'2-Match'} = { 'Match' => { 'Subject' => 'SomeNumber:(\\d\\d\\d\\d)' }, 'Module' => 'Kernel::System::PostMaster::Filter::Match', 'Set' => { 'X-OTRS-TicketKey-1' => 'SomeNumber', 'X-OTRS-TicketValue-1' => '[***]' }, 'StopAfterMatch' => '0' }; </pre>

B.2.4.21. PostMaster::PreFilterModule###3-NewTicketReject

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

Description	Value
Description:	Blocks all the incoming emails that do not have a valid ticket number in subject with From: @example.com address.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'PostMaster::PreFilterModule'}->{'3-NewTicketReject'} = { 'Match' => { 'From' => '@example.com' }, 'Module' => 'Kernel::System::PostMaster::Filter::NewTicketReject', 'Set' => { 'X-OTRS-Ignore' => 'yes' }, 'StopAfterMatch' => '0' }; </pre>

B.2.4.22. PostMaster::PreFilterModule::NewTicketReject::Subject

Description	Value
Description:	Defines the subject for rejected emails.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'PostMaster::PreFilterModule::NewTicketReject::Subject'} = 'Em </pre>

B.2.4.23. PostMaster::PreFilterModule::NewTicketReject::Body

Description	Value
Description:	Defines the body text for rejected emails.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'PostMaster::PreFilterModule::NewTicketReject::Body'} = ' Dear Customer, Unfortunately we could not detect a valid ticket number in your subject, so this email can\'t be processed. Please create a new ticket via the customer panel. Thanks for your help! Your Helpdesk Team ' ; </pre>

B.2.4.24. PostMaster::PreFilterModule###4-CMD

Description	Value
Description:	CMD example setup. Ignores emails where external CMD returns some output on STDOUT (email will be piped into STDIN of some.bin).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'PostMaster::PreFilterModule'}->{'4-CMD'} = { 'CMD' => '/usr/bin/some.bin', 'Module' => 'Kernel::System::PostMaster::Filter::CMD', 'Set' => { 'X-OTRS-Ignore' => 'yes' } };</pre>

B.2.4.25. PostMaster::PreFilterModule###5-SpamAssassin

Description	Value
Description:	Spam Assassin example setup. Ignores emails that are marked with SpamAssassin.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'PostMaster::PreFilterModule'}->{'5-SpamAssassin'} = { 'CMD' => '/usr/bin/spamassassin grep -i "X-Spam-Status: yes"', 'Module' => 'Kernel::System::PostMaster::Filter::CMD', 'Set' => { 'X-OTRS-Ignore' => 'yes' } };</pre>

B.2.4.26. PostMaster::PreFilterModule###6-SpamAssassin

Description	Value
Description:	Spam Assassin example setup. Moves marked mails to spam queue.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'PostMaster::PreFilterModule'}->{'6-SpamAssassin'} = { 'CMD' => '/usr/bin/spamassassin grep -i "X-Spam-Status: yes"', 'Module' => 'Kernel::System::PostMaster::Filter::CMD', 'Set' => { 'X-OTRS-Queue' => 'spam' } };</pre>

B.2.4.27. PostMaster::PreFilterModule###000-MatchDBSource

Description	Value
Description:	Module to use database filter storage.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'PostMaster::PreFilterModule'}->{'000-MatchDBSource'} = { 'Module' => 'Kernel::System::PostMaster::Filter::MatchDBSource' };</pre>

B.2.4.28. PostMaster::PostFilterModule###000-FollowUpArticleTypeCheck

Description	Value
Description:	Module to check if arrived emails should be marked as email-internal (because of original forwarded internal email it college). ArticleType and SenderType define the values for the arrived email/article.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PostMaster::PostFilterModule'}->{'000-FollowUpArticleTypeCheck'} = { 'ArticleType' => 'email-internal', 'Module' => 'Kernel::System::PostMaster::Filter::FollowUpArticleTypeCheck', 'SenderType' => 'customer' };</pre>

B.2.4.29. SendNoAutoResponseRegExp

Description	Value
Description:	If this regex matches, no message will be send by the autoresponder.

Description	Value
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SendNoAutoResponseRegExp'} = ' (MAILER-DAEMON postmaster abuse</code>

B.2.5. Core::Stats

B.2.5.1. Stats::DynamicObjectRegistration###Ticket

Description	Value
Description:	Module to genetare ticket statistics.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Stats::DynamicObjectRegistration'}->{'Ticket'} = { 'Module' => 'Kernel::System::Stats::Dynamic::Ticket' };</code>

B.2.5.2. Stats::DynamicObjectRegistration###TicketList

Description	Value
Description:	Determines if the statistics module may generate ticket lists.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Stats::DynamicObjectRegistration'}->{'TicketList'} = { 'Module' => 'Kernel::System::Stats::Dynamic::TicketList' };</code>

B.2.5.3. Stats::DynamicObjectRegistration###TicketAccountedTime

Description	Value
Description:	Module to genetare accounted time ticket statistics.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Stats::DynamicObjectRegistration'}->{'TicketAccountedTime'} = 'Module' => 'Kernel::System::Stats::Dynamic::TicketAccountedTime' };</pre>

B.2.5.4.**Stats::DynamicObjectRegistration###TicketSolutionResponseTime**

Description	Value
Description:	Module to genetare ticket solution and response time statistics.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Stats::DynamicObjectRegistration'}->{'TicketSolutionResponseTime'} = 'Module' => 'Kernel::System::Stats::Dynamic::TicketSolutionResponseTime' };</pre>

B.2.6. Core::Ticket**B.2.6.1. Ticket::Hook**

Description	Value
Description:	The identifier for a ticket, e.g. Ticket#, Call#, MyTicket#. The default is Ticket#.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Hook'} = 'Ticket#';</pre>

B.2.6.2. 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:	<code>\$Self->{'Ticket::HookDivider'} = " ";</code>

B.2.6.3. Ticket::SubjectSize

Description	Value
Description:	Max size of the subjects in an email reply.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::SubjectSize'} = '100';</code>

B.2.6.4. Ticket::SubjectRe

Description	Value
Description:	The text at the beginning of the subject in an email reply, e.g. RE, AW, or AS.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::SubjectRe'} = 'Re';</code>

B.2.6.5. Ticket::SubjectFwd

Description	Value
Description:	The text at the beginning of the subject when an email is forwarded, e.g. FW, Fwd, or WG.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::SubjectFwd'} = 'Fwd';</code>

B.2.6.6. Ticket::SubjectFormat

Description	Value
Description:	The format of the subject. 'Left' means '[TicketHook#:12345] Some Subject', 'Right' means 'Some Subject [TicketHook#:12345]', 'None' means 'Some Subject' and no ticket number. In the last case you should enable PostmasterFollowupSearchInRaw or PostmasterFollowupSearchInReferences to recognize followups based on email headers and/or body.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::SubjectFormat'} = 'Left';</code>

B.2.6.7. Ticket::CustomQueue

Description	Value
Description:	Name of custom queue. The custom queue is a queue selection of your preferred queues and can be selected in the preferences settings.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::CustomQueue'} = 'My Queues';</code>

B.2.6.8. Ticket::NewArticleIgnoreSystemSender

Description	Value
Description:	Ignore article with system sender type for new article feature (e. g. auto responses or email notifications).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

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

B.2.6.9. Ticket::ChangeOwnerToEveryone

Description	Value
Description:	Changes the owner of tickets to everyone (useful for ASP). Normally only agent with rw permissions in the queue of the ticket will be shown.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::ChangeOwnerToEveryone'} = '0';</code>

B.2.6.10. Ticket::Responsible

Description	Value
Description:	Enables ticket responsible feature, to keep track of a specific ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Responsible'} = '0';</code>

B.2.6.11. Ticket::ResponsibleAutoSet

Description	Value
Description:	Automatically sets the owner of a ticket as the responsible for it (if ticket responsible feature is enabled).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::ResponsibleAutoSet'} = '1';</code>

B.2.6.12. Ticket::Type

Description	Value
Description:	Allows defining new types for ticket (if ticket type feature is enabled).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Type'} = '0';</code>

B.2.6.13. Ticket::Service

Description	Value
Description:	Allows defining services and SLAs for tickets (e. g. email, desktop, network, ...), and escalation attributes for SLAs (if ticket service/SLA feature is enabled).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Service'} = '0';</code>

B.2.6.14. Ticket::ArchiveSystem

Description	Value
Description:	Activates the ticket archive system to have a faster system by moving some tickets out of the daily scope. To search for these tickets, the archive flag has to be enabled in the ticket search.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::ArchiveSystem'} = '0';</code>

B.2.6.15. Ticket::NumberGenerator

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

Description	Value
Description:	Selects the ticket number generator module. "AutoIncrement" increments the ticket number, the SystemID and the counter are used with SystemID.counter format (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 checksum to the string of date and SystemID. The checksum will be rotated on a daily basis. The format looks like Year.Month.Day.SystemID.Counter.CheckSum (e.g. 2002070110101520, 2002070110101535). "Random" generates randomized ticket numbers in the format "SystemID.Random" (e.g. 100057866352, 103745394596).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::NumberGenerator'} = 'Kernel::System::Ticket::Number::</code>

B.2.6.16. Ticket::NumberGenerator::MinCounterSize

Description	Value
Description:	Sets the minimal ticket counter size (if "AutoIncrement" was selected as TicketNumberGenerator). Default is 5, this means the counter starts from 10000.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::NumberGenerator::MinCounterSize'} = '5';</code>

B.2.6.17. Ticket::NumberGenerator::CheckSystemID

Description	Value
Description:	Checks the SystemID in ticket number detection for follow-ups (use "No" if SystemID has been changed after using the system).
Group:	Ticket
SubGroup:	Core::Ticket

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

B.2.6.18. Ticket::CounterLog

Description	Value
Description:	Log file for the ticket counter.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::CounterLog'} = '<OTRS_CONFIG_Home>/var/log/TicketCounter';</code>

B.2.6.19. Ticket::IndexModule

Description	Value
Description:	IndexAccelerator: to choose your backend TicketViewAccelerator module. "RuntimeDB" generates each queue view on the fly from ticket table (no performance problems up to approx. 60.000 tickets in total and 6.000 open tickets in the system). "StaticDB" is the most powerful module, it uses an extra ticket-index table that works like a view (recommended if more than 80.000 and 6.000 open tickets are stored in the system). Use the script "bin/otrs.RebuildTicketIndex.pl" for initial index update.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::IndexModule'} = 'Kernel::System::Ticket::IndexAccelerator';</code>

B.2.6.20. Ticket::StorageModule

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

Description	Value
Description:	Saves the attachments of articles. "DB" stores all data in the database (not recommended for storing big attachments). "FS" stores the data on the filesystem; this is faster but the webserver should run under the OTRS user. You can switch between the modules even on a system that is already in production without any loss of data.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::StorageModule'} = 'Kernel::System::Ticket::ArticleSto</code>

B.2.6.21. ArticleDir

Description	Value
Description:	Specifies the directory to store the data in, if "FS" was selected for TicketStorageModule.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'ArticleDir'} = '<OTRS_CONFIG_Home>/var/article';</code>

B.2.6.22. Ticket::EventManagerPost###100-ArchiveRestore

Description	Value
Description:	Restores a ticket from the archive (only if the event is a state change, from closed to any open available state).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::EventManagerPost'}->{'100-ArchiveRestore'} = { 'Event' => '(TicketStateUpdate)', 'Module' => 'Kernel::System::Ticket::Event::ArchiveRestore', };</code>

B.2.6.23. Ticket::EventModulePost###110-AcceleratorUpdate

Description	Value
Description:	Updates the ticket index accelerator.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}->{'110-AcceleratorUpdate'} = { 'Event' => '(TicketStateUpdate TicketQueueUpdate TicketLockUpdate)', 'Module' => 'Kernel::System::Ticket::Event::TicketAcceleratorUpdate' };</pre>

B.2.6.24. Ticket::EventModulePost###120-ForceOwnerResetOnMove

Description	Value
Description:	Resets and unlocks the owner of a ticket if it was moved to another queue.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}->{'120-ForceOwnerResetOnMove'} = { 'Event' => 'TicketQueueUpdate', 'Module' => 'Kernel::System::Ticket::Event::ForceOwnerReset' };</pre>

B.2.6.25. Ticket::EventModulePost###130-ForceStateChangeOnLock

Description	Value
Description:	Forces to choose a different ticket state (from current) after lock action. Define the current state as key, and the next state after lock action as content.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}->{'130-ForceStateChangeOnLock'} = { 'Event' => 'TicketLockUpdate', 'Module' => 'Kernel::System::Ticket::Event::ForceState', 'new' => 'open' }; </pre>

B.2.6.26. Ticket::EventModulePost###140-ResponsibleAutoSet

Description	Value
Description:	Automatically sets the responsible of a ticket (if it is not set yet) after the first owner update.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}->{'140-ResponsibleAutoSet'} = { 'Event' => 'TicketOwnerUpdate', 'Module' => 'Kernel::System::Ticket::Event::ResponsibleAutoSet' }; </pre>

B.2.6.27. Ticket::EventModulePost###150-TicketPendingTimeReset

Description	Value
Description:	Sets the PendingTime of a ticket to 0 if the state is changed to a non-pending state.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}->{'150-TicketPendingTimeReset'} = { 'Event' => 'TicketStateUpdate', 'Module' => 'Kernel::System::Ticket::Event::TicketPendingTimeReset' }; </pre>

B.2.6.28. Ticket::EventModulePost###500-NotificationEvent

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

Description	Value
Description:	Sends the notifications which are configured in the admin interface under "Notification (Event)".
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}->{'500-NotificationEvent'} = { 'Event' => "", 'Module' => 'Kernel::System::Ticket::Event::NotificationEvent', 'Transaction' => '1' }; </pre>

B.2.6.29. Ticket::EventModulePost###900-EscalationIndex

Description	Value
Description:	Updates the ticket escalation index after a ticket attribute got updated.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}->{'900-EscalationIndex'} = { 'Event' => '(TicketSLAUpdate TicketQueueUpdate TicketStateUpdate Ticket 'Module' => 'Kernel::System::Ticket::Event::TicketEscalationIndex' }; </pre>

B.2.6.30. Ticket::EventModulePost###910-ForceUnlockOnMove

Description	Value
Description:	Forces to unlock tickets after being moved to another queue.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}->{'910-ForceUnlockOnMove'} = { 'Event' => 'TicketQueueUpdate', 'Module' => 'Kernel::System::Ticket::Event::ForceUnlock' }; </pre>

B.2.6.31.**Ticket::EventModulePost###920-TicketArticleNewMessageUpdate**

Description	Value
Description:	Update Ticket "Seen" flag if every article got seen or a new Article got created.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}->{'920-TicketArticleNewMessageUpdate' 'Event' => '(ArticleCreate ArticleFlagSet)', 'Module' => 'Kernel::System::Ticket::Event::TicketNewMessageUpdate' };</pre>

B.2.6.32. Ticket::CustomModule###001-CustomModule

Description	Value
Description:	Overloads (redefines) existing functions in Kernel::System::Ticket. Used to easily add customizations.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::CustomModule'}->{'001-CustomModule'} = 'Kernel::System::Ticket::CustomModule'</pre>

B.2.6.33. Ticket::ViewableSenderTypes

Description	Value
Description:	Defines the default viewable sender types of a ticket (default: customer).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::ViewableSenderTypes'} = ['\customer\';</pre>

B.2.6.34. Ticket::ViewableLocks

Description	Value
Description:	Defines the viewable locks of a ticket. Default: unlock, tmp_lock.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::ViewableLocks'} = ['\unlock\' , '\tmp_lock\'] ;</code>

B.2.6.35. Ticket::ViewableStateType

Description	Value
Description:	Defines the valid state types for a ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::ViewableStateType'} = ['new', 'open', 'pending reminder', 'pending auto'] ;</code>

B.2.6.36. Ticket::UnlockStateType

Description	Value
Description:	Defines the valid states for unlocked tickets. To unlock tickets the script "bin/otrs.UnlockTickets.pl" can be used.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::UnlockStateType'} = ['new', 'open'] ;</code>

B.2.6.37. Ticket::PendingNotificationOnlyToOwner

Description	Value
Description:	Sends reminder notifications of unlocked ticket after reaching the reminder date (only sent to ticket owner).

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

B.2.6.38. Ticket::PendingNotificationNotToResponsible

Description	Value
Description:	Disables sending reminder notifications to the responsible agent of a ticket (Ticket::Responsible needs to be activated).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::PendingNotificationNotToResponsible'} = '0';</code>

B.2.6.39. Ticket::PendingReminderStateType

Description	Value
Description:	Defines the state type of the reminder for pending tickets.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::PendingReminderStateType'} = ['pending reminder'];</code>

B.2.6.40. Ticket::PendingAutoStateType

Description	Value
Description:	Determines the possible states for pending tickets that changed state after reaching time limit.
Group:	Ticket

Description	Value
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::PendingAutoStateType'} = ['pending auto'];</code>

B.2.6.41. Ticket::StateAfterPending

Description	Value
Description:	Defines which states should be set automatically (Content), after the pending time of state (Key) has been reached.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::StateAfterPending'} = { 'pending auto close+' => 'closed successful', 'pending auto close-' => 'closed unsuccessful' };</code>

B.2.6.42. System::Permission

Description	Value
Description:	Standard available permissions for agents within the application. If more permissions are needed, they can be entered here. Permissions must be defined to be effective. Some other good permissions have also been provided built-in: note, close, pending, customer, freetext, move, compose, responsible, forward, and bounce. Make sure that "rw" is always the last registered permission.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'System::Permission'} = ['ro', 'move_into', 'create', 'note', 'owner', 'priority', 'rw'];</code>

B.2.6.43. Ticket::Permission###1-OwnerCheck

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

Description	Value
Description:	Module to check the owner of a ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Permission'}->{'1-OwnerCheck'} = { 'Granted' => '1', 'Module' => 'Kernel::System::Ticket::Permission::OwnerCheck', 'Required' => '0' }; </pre>

B.2.6.44. Ticket::Permission###2-ResponsibleCheck

Description	Value
Description:	Module to check the agent responsible of a ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Permission'}->{'2-ResponsibleCheck'} = { 'Granted' => '1', 'Module' => 'Kernel::System::Ticket::Permission::ResponsibleCheck', 'Required' => '0' }; </pre>

B.2.6.45. Ticket::Permission###3-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:	0
Config-Setting:	<pre>\$Self->{'Ticket::Permission'}->{'3-GroupCheck'} = { 'Granted' => '1', 'Module' => 'Kernel::System::Ticket::Permission::GroupCheck', 'Required' => '0' }; </pre>

B.2.6.46. Ticket::Permission###4-WatcherCheck

Description	Value
Description:	Module to check the watcher agents of a ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Permission'}->{'4-WatcherCheck'} = { 'Granted' => '1', 'Module' => 'Kernel::System::Ticket::Permission::WatcherCheck', 'Required' => '0' }; </pre>

B.2.6.47. CustomerTicket::Permission###1-GroupCheck

Description	Value
Description:	Module to check the group permissions for the access to customer tickets.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerTicket::Permission'}->{'1-GroupCheck'} = { 'Granted' => '0', 'Module' => 'Kernel::System::Ticket::CustomerPermission::GroupCheck', 'Required' => '1' }; </pre>

B.2.6.48. CustomerTicket::Permission###2-CustomerUserIDCheck

Description	Value
Description:	Grants access, if the customer ID of the ticket matches the customer user's ID and the customer user has group permissions on the queue the ticket is in.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerTicket::Permission'}->{'2-CustomerUserIDCheck'} = { 'Granted' => '1', 'Module' => 'Kernel::System::Ticket::CustomerPermission::CustomerUserIDCheck', 'Required' => '0' }; </pre>

B.2.6.49. CustomerTicket::Permission###3-CustomerIDCheck

Description	Value
Description:	Module to check customer permissions.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerTicket::Permission'}->{'3-CustomerIDCheck'} = { 'Granted' => '1', 'Module' => 'Kernel::System::Ticket::CustomerPermission::CustomerIDCheck', 'Required' => '0' }; </pre>

B.2.6.50. Ticket::DefineEmailFrom

Description	Value
Description:	Defines how the From field from the emails (sent from answers and email tickets) should look like.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::DefineEmailFrom'} = 'SystemAddressName'; </pre>

B.2.6.51. Ticket::DefineEmailFromSeparator

Description	Value
Description:	Defines the separator between the agents real name and the given queue email address.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::DefineEmailFromSeparator'} = 'via'; </pre>

B.2.6.52. CustomerNotifyJustToRealCustomer

Description	Value
Description:	Sends customer notifications just to the mapped customer. Normally, if no customer is mapped, the latest customer sender gets the notification.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerNotifyJustToRealCustomer'} = '0';</code>

B.2.6.53. AgentSelfNotifyOnAction

Description	Value
Description:	Specifies if an agent should receive email notification of his own actions.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'AgentSelfNotifyOnAction'} = '0';</code>

B.2.7. Core::TicketACL**B.2.7.1. Ticket::Acl::Module###1-Ticket::Acl::Module**

Description	Value
Description:	ACL module that allows closing parent tickets only if all its children are already closed ("State" shows which states are not available for the parent ticket until all child tickets are closed).
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Acl::Module'}->{'1-Ticket::Acl::Module'} = { 'Module' => 'Kernel::System::Ticket::Acl::CloseParentAfterClosedChilds', 'State' => ['closed successful', 'closed unsuccessful'] };</code>

B.2.7.2. TicketACL::Default::Action

Description	Value
Description:	Default ACL values for ticket actions.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketACL::Default::Action'} = {};</code>

B.2.8. Core::TicketBulkAction**B.2.8.1. Ticket::Frontend::BulkFeature**

Description	Value
Description:	Enables ticket bulk action feature for the agent frontend to work on more than one ticket at a time.
Group:	Ticket
SubGroup:	Core::TicketBulkAction
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::BulkFeature'} = '1';</code>

B.2.8.2. Ticket::Frontend::BulkFeatureGroup

Description	Value
Description:	Enables ticket bulk action feature only for the listed groups.
Group:	Ticket
SubGroup:	Core::TicketBulkAction
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::BulkFeatureGroup'} = ['admin', 'users'];</code>

B.2.9. Core::TicketFreeText

B.2.9.1. TicketFreeKey1

Description	Value
Description:	Defines the free key field number 1 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey1'} = { 'Product' => 'Product' };</code>

B.2.9.2. TicketFreeKey1::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 1 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey1::DefaultSelection'} = "";</code>

B.2.9.3. TicketFreeText1

Description	Value
Description:	Defines the free text field number 1 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText1'} = { '' => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

B.2.9.4. TicketFreeText1::DefaultSelection

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

Description	Value
Description:	Defines the default selection of the free text field number 1 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText1::DefaultSelection'} = 'Notebook';</code>

B.2.9.5. TicketFreeText1::Link

Description	Value
Description:	Defines the http link for the free text field number 1 for tickets (it will be used in every ticket view).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText1::Link'} = 'http://some.example.com/handle?que</code>

B.2.9.6. TicketFreeKey2

Description	Value
Description:	Defines the free key field number 2 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey2'} = { 'Product' => 'Product' };</code>

B.2.9.7. TicketFreeKey2::DefaultSelection

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

Description	Value
Description:	Defines the default selection of the free key field number 2 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey2::DefaultSelection'} = '';</code>

B.2.9.8. TicketFreeText2

Description	Value
Description:	Defines the free text field number 2 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText2'} = { '' => '- ', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

B.2.9.9. TicketFreeText2::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 2 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText2::DefaultSelection'} = 'Notebook';</code>

B.2.9.10. TicketFreeText2::Link

Description	Value
Description:	Defines the http link for the free text field number 2 for tickets (it will be used in every ticket view).

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText2::Link'} = 'http://some.example.com/handle?que</code>

B.2.9.11. TicketFreeKey3

Description	Value
Description:	Defines the free key field number 3 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey3'} = { 'Product' => 'Product' };</code>

B.2.9.12. TicketFreeKey3::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 3 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey3::DefaultSelection'} = '';</code>

B.2.9.13. TicketFreeText3

Description	Value
Description:	Defines the free text field number 3 for ticket to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'TicketFreeText3'} = { " => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' }; </pre>

B.2.9.14. TicketFreeText3::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 3 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText3::DefaultSelection'} = 'Notebook'; </pre>

B.2.9.15. TicketFreeText3::Link

Description	Value
Description:	Defines the http link for the free text field number 3 for tickets (it will be used in every ticket view).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText3::Link'} = 'http://some.example.com/handle?que </pre>

B.2.9.16. TicketFreeKey4

Description	Value
Description:	Defines the free key field number 4 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'TicketFreeKey4'} = { 'Product' => 'Product' };</code>

B.2.9.17. TicketFreeKey4::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 4 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey4::DefaultSelection'} = '';</code>

B.2.9.18. TicketFreeText4

Description	Value
Description:	Defines the free text field number 4 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText4'} = { '' => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

B.2.9.19. TicketFreeText4::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 4 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText4::DefaultSelection'} = 'Notebook';</code>

B.2.9.20. TicketFreeText4::Link

Description	Value
Description:	Defines the http link for the free text field number 4 for tickets (it will be used in every ticket view).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText4::Link'} = 'http://some.example.com/handle?que</code>

B.2.9.21. TicketFreeKey5

Description	Value
Description:	Defines the free key field number 5 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey5'} = { 'Product' => 'Product' };</code>

B.2.9.22. TicketFreeKey5::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 5 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey5::DefaultSelection'} = '';</code>

B.2.9.23. TicketFreeText5

Description	Value
Description:	Defines the free text field number 5 for tickets to add a new ticket attribute.

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText5'} = { 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

B.2.9.24. TicketFreeText5::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 5 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText5::DefaultSelection'} = 'Notebook';</pre>

B.2.9.25. TicketFreeText5::Link

Description	Value
Description:	Defines the http link for the free text field number 5 for tickets (it will be used in every ticket view).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText5::Link'} = 'http://some.example.com/handle?que</pre>

B.2.9.26. TicketFreeKey6

Description	Value
Description:	Defines the free key field number 6 for tickets to add a new ticket attribute.

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey6'} = { 'Product' => 'Product' };</code>

B.2.9.27. TicketFreeKey6::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 6 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey6::DefaultSelection'} = "";</code>

B.2.9.28. TicketFreeText6

Description	Value
Description:	Defines the free text field number 6 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText6'} = { "" => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

B.2.9.29. TicketFreeText6::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 6 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0

Description	Value
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText6::DefaultSelection'} = 'Notebook';</code>

B.2.9.30. TicketFreeText6::Link

Description	Value
Description:	Defines the http link for the free text field number 6 for tickets (it will be used in every ticket view).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText6::Link'} = 'http://some.example.com/handle?que</code>

B.2.9.31. TicketFreeKey7

Description	Value
Description:	Defines the free key field number 7 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey7'} = { 'Product' => 'Product' };</code>

B.2.9.32. TicketFreeKey7::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 7 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'TicketFreeKey7::DefaultSelection'} = '';</code>

B.2.9.33. TicketFreeText7

Description	Value
Description:	Defines the free text field number 7 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText7'} = { '' => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

B.2.9.34. TicketFreeText7::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 7 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText7::DefaultSelection'} = 'Notebook';</code>

B.2.9.35. TicketFreeText7::Link

Description	Value
Description:	Defines the http link for the free text field number 7 for tickets (it will be used in every ticket view).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'TicketFreeText7::Link'} = 'http://some.example.com/handle?que</code>

B.2.9.36. TicketFreeKey8

Description	Value
Description:	Defines the free key field number 8 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey8'} = { 'Product' => 'Product' };</code>

B.2.9.37. TicketFreeKey8::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 8 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey8::DefaultSelection'} = "";</code>

B.2.9.38. TicketFreeText8

Description	Value
Description:	Defines the free text field number 8 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText8'} = { '' => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

B.2.9.39. TicketFreeText8::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 8 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText8::DefaultSelection'} = 'Notebook';</code>

B.2.9.40. TicketFreeText8::Link

Description	Value
Description:	Defines the http link for the free text field number 8 for tickets (it will be used in every ticket view).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText8::Link'} = 'http://some.example.com/handle?que</code>

B.2.9.41. TicketFreeKey9

Description	Value
Description:	Defines the free key field number 9 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey9'} = { 'Product' => 'Product' };</code>

B.2.9.42. TicketFreeKey9::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 9 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey9::DefaultSelection'} = '';</code>

B.2.9.43. TicketFreeText9

Description	Value
Description:	Defines the free text field number 9 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText9'} = { '' => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

B.2.9.44. TicketFreeText9::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 9 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText9::DefaultSelection'} = 'Notebook';</code>

B.2.9.45. TicketFreeText9::Link

Description	Value
Description:	Defines the http link for the free text field number 9 for tickets (it will be used in every ticket view).

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText9::Link'} = 'http://some.example.com/handle?que</code>

B.2.9.46. TicketFreeKey10

Description	Value
Description:	Defines the free key field number 10 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey10'} = { 'Product' => 'Product' };</code>

B.2.9.47. TicketFreeKey10::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 10 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey10::DefaultSelection'} = "";</code>

B.2.9.48. TicketFreeText10

Description	Value
Description:	Defines the free text field number 10 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'TicketFreeText10'} = { " => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

B.2.9.49. TicketFreeText10::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 10 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText10::DefaultSelection'} = 'Notebook';</pre>

B.2.9.50. TicketFreeText10::Link

Description	Value
Description:	Defines the http link for the free text field number 10 for tickets (it will be used in every ticket view).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText10::Link'} = 'http://some.example.com/handle?qu</pre>

B.2.9.51. TicketFreeKey11

Description	Value
Description:	Defines the free key field number 11 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'TicketFreeKey11'} = { 'Product' => 'Product' };</code>

B.2.9.52. TicketFreeKey11::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 11 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey11::DefaultSelection'} = "";</code>

B.2.9.53. TicketFreeText11

Description	Value
Description:	Defines the free text field number 11 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText11'} = { " => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

B.2.9.54. TicketFreeText11::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 11 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText11::DefaultSelection'} = 'Notebook';</code>

B.2.9.55. TicketFreeText11::Link

Description	Value
Description:	Defines the http link for the free text field number 11 for tickets (it will be used in every ticket view).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText11::Link'} = 'http://some.example.com/handle?qu</code>

B.2.9.56. TicketFreeKey12

Description	Value
Description:	Defines the free key field number 12 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey12'} = { 'Product' => 'Product' };</code>

B.2.9.57. TicketFreeKey12::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 12 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey12::DefaultSelection'} = '';</code>

B.2.9.58. TicketFreeText12

Description	Value
Description:	Defines the free text field number 12 for tickets to add a new ticket attribute.

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText12'} = { ' ' => '- ', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

B.2.9.59. TicketFreeText12::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 12 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText12::DefaultSelection'} = 'Notebook';</code>

B.2.9.60. TicketFreeText12::Link

Description	Value
Description:	Defines the http link for the free text field number 12 for tickets (it will be used in every ticket view).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText12::Link'} = 'http://some.example.com/handle?qu</code>

B.2.9.61. TicketFreeKey13

Description	Value
Description:	Defines the free key field number 13 for tickets to add a new ticket attribute.

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey13'} = { 'Product' => 'Product' };</code>

B.2.9.62. TicketFreeKey13::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 13 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey13::DefaultSelection'} = "";</code>

B.2.9.63. TicketFreeText13

Description	Value
Description:	Defines the free text field number 13 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText13'} = { " => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

B.2.9.64. TicketFreeText13::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 13 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0

Description	Value
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText13::DefaultSelection'} = 'Notebook';</code>

B.2.9.65. TicketFreeText13::Link

Description	Value
Description:	Defines the http link for the free text field number 13 for tickets (it will be used in every ticket view).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText13::Link'} = 'http://some.example.com/handle?qu</code>

B.2.9.66. TicketFreeKey14

Description	Value
Description:	Defines the free key field number 14 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey14'} = { 'Product' => 'Product' };</code>

B.2.9.67. TicketFreeKey14::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 14 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'TicketFreeKey14::DefaultSelection'} = '';</code>

B.2.9.68. TicketFreeText14

Description	Value
Description:	Defines the free text field number 14 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText14'} = { '' => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

B.2.9.69. TicketFreeText14::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 14 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText14::DefaultSelection'} = 'Notebook';</code>

B.2.9.70. TicketFreeText14::Link

Description	Value
Description:	Defines the http link for the free text field number 14 for tickets (it will be used in every ticket view).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'TicketFreeText14::Link'} = 'http://some.example.com/handle?qu</code>

B.2.9.71. TicketFreeKey15

Description	Value
Description:	Defines the free key field number 15 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey15'} = { 'Product' => 'Product' };</code>

B.2.9.72. TicketFreeKey15::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 15 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey15::DefaultSelection'} = "";</code>

B.2.9.73. TicketFreeText15

Description	Value
Description:	Defines the free text field number 15 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText15'} = { "" => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

B.2.9.74. TicketFreeText15::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 15 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText15::DefaultSelection'} = 'Notebook';</code>

B.2.9.75. TicketFreeText15::Link

Description	Value
Description:	Defines the http link for the free text field number 15 for tickets (it will be used in every ticket view).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText15::Link'} = 'http://some.example.com/handle?qu</code>

B.2.9.76. TicketFreeKey16

Description	Value
Description:	Defines the free key field number 16 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey16'} = { 'Product' => 'Product' };</code>

B.2.9.77. TicketFreeKey16::DefaultSelection

Description	Value
Description:	Defines the default selection of the free key field number 16 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey16::DefaultSelection'} = '';</code>

B.2.9.78. TicketFreeText16

Description	Value
Description:	Defines the free text field number 16 for tickets to add a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText16'} = { '' => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

B.2.9.79. TicketFreeText16::DefaultSelection

Description	Value
Description:	Defines the default selection of the free text field number 16 for tickets (if more than one option is provided).
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText16::DefaultSelection'} = 'Notebook';</code>

B.2.9.80. TicketFreeText16::Link

Description	Value
Description:	Defines the http link for the free text field number 16 for tickets (it will be used in every ticket view).

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText16::Link'} = 'http://some.example.com/handle?qu</code>

B.2.10. Core::TicketFreeTextDefault

B.2.10.1. Ticket::EventModulePost###TicketFreeFieldDefault

Description	Value
Description:	Event module registration. For more performance you can define a trigger event (e. g. Event => TicketCreate). This is only possible if all TicketFreeField elements need the same event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::EventModulePost'}->{'TicketFreeFieldDefault'} = { 'Module' => 'Kernel::System::Ticket::Event::TicketFreeFieldDefault', 'Transaction' => '1' };</code>

B.2.10.2. Ticket::TicketFreeFieldDefault###Element1

Description	Value
Description:	Configures a default TicketFreeField setting. "Counter" defines the free text field which should be used, "Key" is the TicketFreeKey, "Value" is the TicketFreeText and "Event" defines the trigger event. Please check the developer manual (http://doc.otsr.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}->{'Element1'} = { 'Counter' => '1', 'Event' => 'TicketCreate', 'Key' => 'TicketType', 'Value' => 'Default' }; </pre>

B.2.10.3. Ticket::TicketFreeFieldDefault###Element2

Description	Value
Description:	Configures a default TicketFreeField setting. "Counter" defines the free text field which should be used, "Key" is the TicketFreeKey, "Value" is the TicketFreeText and "Event" defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}->{'Element2'} = { 'Counter' => "", 'Event' => "", 'Key' => "", 'Value' => "" }; </pre>

B.2.10.4. Ticket::TicketFreeFieldDefault###Element3

Description	Value
Description:	Configures a default TicketFreeField setting. "Counter" defines the free text field which should be used, "Key" is the TicketFreeKey, "Value" is the TicketFreeText and "Event" defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}->{'Element3'} = { 'Counter' => "", 'Event' => "", 'Key' => "", 'Value' => "" }; </pre>

B.2.10.5. Ticket::TicketFreeFieldDefault###Element4

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

Description	Value
Description:	Configures a default TicketFreeField setting. "Counter" defines the free text field which should be used, "Key" is the TicketFreeKey, "Value" is the TicketFreeText and "Event" defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}->{'Element4'} = { 'Counter' => "", 'Event' => "", 'Key' => "", 'Value' => "" };</pre>

B.2.10.6. Ticket::TicketFreeFieldDefault###Element5

Description	Value
Description:	Configures a default TicketFreeField setting. "Counter" defines the free text field which should be used, "Key" is the TicketFreeKey, "Value" is the TicketFreeText and "Event" defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}->{'Element5'} = { 'Counter' => "", 'Event' => "", 'Key' => "", 'Value' => "" };</pre>

B.2.10.7. Ticket::TicketFreeFieldDefault###Element6

Description	Value
Description:	Configures a default TicketFreeField setting. "Counter" defines the free text field which should be used, "Key" is the TicketFreeKey, "Value" is the TicketFreeText and "Event" defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}->{'Element6'} = { 'Counter' => "", 'Event' => "", 'Key' => "", 'Value' => " };</pre>

B.2.10.8. Ticket::TicketFreeFieldDefault###Element7

Description	Value
Description:	Configures a default TicketFreeField setting. "Counter" defines the free text field which should be used, "Key" is the TicketFreeKey, "Value" is the TicketFreeText and "Event" defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}->{'Element7'} = { 'Counter' => "", 'Event' => "", 'Key' => "", 'Value' => " };</pre>

B.2.10.9. Ticket::TicketFreeFieldDefault###Element8

Description	Value
Description:	Configures a default TicketFreeField setting. "Counter" defines the free text field which should be used, "Key" is the TicketFreeKey, "Value" is the TicketFreeText and "Event" defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}->{'Element8'} = { 'Counter' => "", 'Event' => "", 'Key' => "", 'Value' => " };</pre>

B.2.10.10. Ticket::TicketFreeFieldDefault###Element9

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

Description	Value
Description:	Configures a default TicketFreeField setting. "Counter" defines the free text field which should be used, "Key" is the TicketFreeKey, "Value" is the TicketFreeText and "Event" defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}->{'Element9'} = { 'Counter' => "", 'Event' => "", 'Key' => "", 'Value' => "" };</pre>

B.2.10.11. Ticket::TicketFreeFieldDefault###Element10

Description	Value
Description:	Configures a default TicketFreeField setting. "Counter" defines the free text field which should be used, "Key" is the TicketFreeKey, "Value" is the TicketFreeText and "Event" defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}->{'Element10'} = { 'Counter' => "", 'Event' => "", 'Key' => "", 'Value' => "" };</pre>

B.2.10.12. Ticket::TicketFreeFieldDefault###Element11

Description	Value
Description:	Configures a default TicketFreeField setting. "Counter" defines the free text field which should be used, "Key" is the TicketFreeKey, "Value" is the TicketFreeText and "Event" defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}->{'Element11'} = { 'Counter' => "", 'Event' => "", 'Key' => "", 'Value' => " "};</pre>

B.2.10.13. Ticket::TicketFreeFieldDefault###Element12

Description	Value
Description:	Configures a default TicketFreeField setting. "Counter" defines the free text field which should be used, "Key" is the TicketFreeKey, "Value" is the TicketFreeText and "Event" defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}->{'Element12'} = { 'Counter' => "", 'Event' => "", 'Key' => "", 'Value' => " "};</pre>

B.2.10.14. Ticket::TicketFreeFieldDefault###Element13

Description	Value
Description:	Configures a default TicketFreeField setting. "Counter" defines the free text field which should be used, "Key" is the TicketFreeKey, "Value" is the TicketFreeText and "Event" defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}->{'Element13'} = { 'Counter' => "", 'Event' => "", 'Key' => "", 'Value' => " "};</pre>

B.2.10.15. Ticket::TicketFreeFieldDefault###Element14

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

Description	Value
Description:	Configures a default TicketFreeField setting. "Counter" defines the free text field which should be used, "Key" is the TicketFreeKey, "Value" is the TicketFreeText and "Event" defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}->{'Element14'} = { 'Counter' => "", 'Event' => "", 'Key' => "", 'Value' => "" }; </pre>

B.2.10.16. Ticket::TicketFreeFieldDefault###Element15

Description	Value
Description:	Configures a default TicketFreeField setting. "Counter" defines the free text field which should be used, "Key" is the TicketFreeKey, "Value" is the TicketFreeText and "Event" defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}->{'Element15'} = { 'Counter' => "", 'Event' => "", 'Key' => "", 'Value' => "" }; </pre>

B.2.10.17. Ticket::TicketFreeFieldDefault###Element16

Description	Value
Description:	Configures a default TicketFreeField setting. "Counter" defines the free text field which should be used, "Key" is the TicketFreeKey, "Value" is the TicketFreeText and "Event" defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}->{'Element16'} = { 'Counter' => "", 'Event' => "", 'Key' => "", 'Value' => " };</pre>

B.2.11. Core::TicketFreeTime

B.2.11.1. TicketFreeTimeKey1

Description	Value
Description:	Defines the free time key field number 1 for tickets.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TicketFreeTimeKey1'} = 'Time1';</pre>

B.2.11.2. TicketFreeTimeOptional1

Description	Value
Description:	Defines whether the free time field number 1 is optional or not.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TicketFreeTimeOptional1'} = '1';</pre>

B.2.11.3. TicketFreeTimeDiff1

Description	Value
Description:	Defines the difference from now (in seconds) of the free time field number 1's default value.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'TicketFreeTimeDiff1'} = '0';</code>

B.2.11.4. TicketFreeTimePeriod1

Description	Value
Description:	Defines the years (in future and in past) which can get selected in free time field number 1.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimePeriod1'} = { 'YearPeriodFuture' => '5', 'YearPeriodPast' => '5' };</code>

B.2.11.5. TicketFreeTimeKey2

Description	Value
Description:	Defines the free time key field number 2 for tickets.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeKey2'} = 'Time2';</code>

B.2.11.6. TicketFreeTimeOptional2

Description	Value
Description:	Defines whether the free time field number 2 is optional or not.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeOptional2'} = '1';</code>

B.2.11.7. TicketFreeTimeDiff2

Description	Value
Description:	Defines the difference from now (in seconds) of the free time field number 2's default value.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimeDiff2'} = '0';</code>

B.2.11.8. TicketFreeTimePeriod2

Description	Value
Description:	Defines the years (in future and in past) which can get selected in free time field number 2.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimePeriod2'} = { 'YearPeriodFuture' => '5', 'YearPeriodPast' => '5'};</code>

B.2.11.9. TicketFreeTimeKey3

Description	Value
Description:	Defines the free time key field number 3 for tickets.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeKey3'} = 'Time3';</code>

B.2.11.10. TicketFreeTimeOptional3

Description	Value
Description:	Defines whether the free time field number 3 is optional or not.
Group:	Ticket
SubGroup:	Core::TicketFreeTime

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeOptional3'} = '1';</code>

B.2.11.11. TicketFreeTimeDiff3

Description	Value
Description:	Defines the difference from now (in seconds) of the free time field number 3's default value.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimeDiff3'} = '0';</code>

B.2.11.12. TicketFreeTimePeriod3

Description	Value
Description:	Defines the years (in future and in past) which can get selected in free time field number 3.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimePeriod3'} = { 'YearPeriodFuture' => '5', 'YearPeriodPast' => '5' };</code>

B.2.11.13. TicketFreeTimeKey4

Description	Value
Description:	Defines the free time key field number 4 for tickets.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeKey4'} = 'Time4';</code>

B.2.11.14. TicketFreeTimeOptional4

Description	Value
Description:	Defines whether the free time field number 4 is optional or not.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeOptional4'} = '1';</code>

B.2.11.15. TicketFreeTimeDiff4

Description	Value
Description:	Defines the difference from now (in seconds) of the free time field number 4's default value.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimeDiff4'} = '0';</code>

B.2.11.16. TicketFreeTimePeriod4

Description	Value
Description:	Defines the years (in future and in past) which can get selected in free time field number 4.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimePeriod4'} = { 'YearPeriodFuture' => '5', 'YearPeriodPast' => '5' };</code>

B.2.11.17. TicketFreeTimeKey5

Description	Value
Description:	Defines the free time key field number 5 for tickets.
Group:	Ticket

Description	Value
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeKey5'} = 'Time5';</code>

B.2.11.18. TicketFreeTimeOptional5

Description	Value
Description:	Defines whether the free time field number 5 is optional or not.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeOptional5'} = '1';</code>

B.2.11.19. TicketFreeTimeDiff5

Description	Value
Description:	Defines the difference from now (in seconds) of the free time field number 5's default value.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimeDiff5'} = '0';</code>

B.2.11.20. TicketFreeTimePeriod5

Description	Value
Description:	Defines the years (in future and in past) which can get selected in free time field number 5.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimePeriod5'} = { 'YearPeriodFuture' => '5', 'YearPeriodPast' => '5' };</code>

B.2.11.21. TicketFreeTimeKey6

Description	Value
Description:	Defines the free time key field number 6 for tickets.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeKey6'} = 'Time6';</code>

B.2.11.22. TicketFreeTimeOptional6

Description	Value
Description:	Defines whether the free time field number 6 is optional or not.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeOptional6'} = '1';</code>

B.2.11.23. TicketFreeTimeDiff6

Description	Value
Description:	Defines the difference from now (in seconds) of the free time field number 6's default value.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimeDiff6'} = '0';</code>

B.2.11.24. TicketFreeTimePeriod6

Description	Value
Description:	Defines the years (in future and in past) which can get selected in free time field number 6.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0

Description	Value
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimePeriod6'} = { 'YearPeriodFuture' => '5', 'YearPeriodPast' => '5' };</code>

B.2.12. Core::TicketWatcher

B.2.12.1. Ticket::Watcher

Description	Value
Description:	Enables or disables the ticket watcher feature, to keep track of tickets without being the owner nor the responsible.
Group:	Ticket
SubGroup:	Core::TicketWatcher
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Watcher'} = '0';</code>

B.2.12.2. Ticket::WatcherGroup

Description	Value
Description:	Enables ticket watcher feature only for the listed groups.
Group:	Ticket
SubGroup:	Core::TicketWatcher
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::WatcherGroup'} = ['admin', 'users'];</code>

B.2.13. Frontend::Admin::ModuleRegistration

B.2.13.1. Frontend::Module###AdminQueue

Description	Value
Description:	Frontend module registration for the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminQueue'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Queue', 'Description' => 'Create and manage queues.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Queues', 'Prio' => '100' }, 'NavBarName' => 'Admin', 'Title' => 'Queue' };</pre>

B.2.13.2. Frontend::Module###AdminResponse

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminResponse'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Queue', 'Description' => 'Create and manage response templates.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Responses', 'Prio' => '200' }, 'NavBarName' => 'Admin', 'Title' => 'Response' };</pre>

B.2.13.3. Frontend::Module###AdminQueueResponses

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminQueueResponses'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Queue', 'Description' => 'Link responses to queues.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Responses <-> Queues', 'Prio' => '300' }, 'NavBarName' => 'Admin', 'Title' => 'Responses <-> Queues' }; </pre>

B.2.13.4. Frontend::Module###AdminAutoResponse

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminAutoResponse'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Queue', 'Description' => 'Create and manage responses that are automatically se 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Auto Responses', 'Prio' => '400' }, 'NavBarName' => 'Admin', 'Title' => 'Auto Responses' }; </pre>

B.2.13.5. Frontend::Module###AdminQueueAutoResponse

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminQueueAutoResponse'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Queue', 'Description' => 'Link queues to auto responses.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Auto Responses <-> Queues', 'Prio' => '500' }, 'NavBarName' => 'Admin', 'Title' => 'Queues <-> Auto Responses' }; </pre>

B.2.13.6. Frontend::Module###AdminAttachment

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminAttachment'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Queue', 'Description' => 'Create and manage attachments.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Attachments', 'Prio' => '600' }, 'NavBarName' => 'Admin', 'Title' => 'Attachment' }; </pre>

B.2.13.7. Frontend::Module###AdminResponseAttachment

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminResponseAttachment'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Queue', 'Description' => 'Link attachments to responses templates.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Attachments <-> Responses', 'Prio' => '700' }, 'NavBarName' => 'Admin', 'Title' => 'Attachments <-> Responses' }; </pre>

B.2.13.8. Frontend::Module###AdminSalutation

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminSalutation'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Queue', 'Description' => 'Create and manage salutations.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Salutations', 'Prio' => '800' }, 'NavBarName' => 'Admin', 'Title' => 'Salutations' }; </pre>

B.2.13.9. Frontend::Module###AdminSignature

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminSignature'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Queue', 'Description' => 'Create and manage signatures.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Signatures', 'Prio' => '900' }, 'NavBarName' => 'Admin', 'Title' => 'Signatures' }; </pre>

B.2.13.10. Frontend::Module###AdminSystemAddress

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminSystemAddress'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Email', 'Description' => 'Set sender email addresses for this system.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Email Addresses', 'Prio' => '300' }, 'NavBarName' => 'Admin', 'Title' => 'System address' };</pre>

B.2.13.11. Frontend::Module###AdminNotification

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminNotification'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Ticket', 'Description' => 'Create and manage notifications that are sent to agent', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Agent Notifications', 'Prio' => '400' }, 'NavBarName' => 'Admin', 'Title' => 'Agent Notifications' };</pre>

B.2.13.12. Frontend::Module###AdminNotificationEvent

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminNotificationEvent'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Ticket', 'Description' => 'Create and manage event based notifications.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Notifications (Event)', 'Prio' => '400' }, 'NavBarName' => 'Admin', 'Title' => 'Notifications (Event)' };</pre>

B.2.13.13. Frontend::Module###AdminService

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminService'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Ticket', 'Description' => 'Create and manage services.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Services', 'Prio' => '900' }, 'NavBarName' => 'Admin', 'Title' => 'Services' };</pre>

B.2.13.14. Frontend::Module###AdminSLA

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminSLA'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Ticket', 'Description' => 'Create and manage Service Level Agreements (SLAs).', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Service Level Agreements', 'Prio' => '1000' }, 'NavBarName' => 'Admin', 'Title' => 'Service Level Agreements' }; </pre>

B.2.13.15. Frontend::Module###AdminType

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminType'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Ticket', 'Description' => 'Create and manage ticket types.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Types', 'Prio' => '700' }, 'NavBarName' => 'Admin', 'Title' => 'Types' }; </pre>

B.2.13.16. Frontend::Module###AdminState

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminState'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Ticket', 'Description' => 'Create and manage ticket states.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'States', 'Prio' => '800' }, 'NavBarName' => 'Admin', 'Title' => 'States' }; </pre>

B.2.13.17. Frontend::Module###AdminPriority

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminPriority'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Ticket', 'Description' => 'Create and manage ticket priorities.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Priorities', 'Prio' => '850' }, 'NavBarName' => 'Admin', 'Title' => 'Priorities' }; </pre>

B.2.13.18. Frontend::Module###AdminGenericAgent

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminGenericAgent'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'System', 'Description' => 'Manage periodic tasks.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'GenericAgent', 'Prio' => '300' }, 'NavBarName' => 'Admin', 'Title' => 'GenericAgent' }; </pre>

B.2.14. Frontend::Agent**B.2.14.1. Ticket::Frontend::PendingDiffTime**

Description	Value
Description:	Time in seconds that gets added to the actual time if setting a pending-state (default: 86400 = 1 day).

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PendingDiffTime'} = '86400';</code>

B.2.14.2. Ticket::Frontend::ListType

Description	Value
Description:	Shows existing parent/child queue lists in the system in the form of a tree or a list.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::ListType'} = 'tree';</code>

B.2.14.3. Ticket::Frontend::TextAreaEmail

Description	Value
Description:	Permitted width for compose email windows.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::TextAreaEmail'} = '82';</code>

B.2.14.4. Ticket::Frontend::TextAreaNote

Description	Value
Description:	Permitted width for compose note windows.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::TextAreaNote'} = '78';</code>

B.2.14.5. Ticket::Frontend::CustomerInfoCompose

Description	Value
Description:	Shows the customer user information (phone and email) in the compose screen.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerInfoCompose'} = '1';</code>

B.2.14.6. Ticket::Frontend::CustomerInfoComposeMaxSize

Description	Value
Description:	Max size (in characters) of the customer information table (phone and email) in the compose screen.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerInfoComposeMaxSize'} = '22';</code>

B.2.14.7. Ticket::Frontend::CustomerInfoZoom

Description	Value
Description:	Shows the customer user's info in the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerInfoZoom'} = '1';</code>

B.2.14.8. Ticket::Frontend::CustomerInfoZoomMaxSize

Description	Value
Description:	Maximum size (in characters) of the customer information table in the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerInfoZoomMaxSize'} = '22';</code>

B.2.14.9. Ticket::Frontend::CustomerInfoQueueMaxSize

Description	Value
Description:	Maximum size (in characters) of the customer info table in the queue view.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerInfoQueueMaxSize'} = '18';</code>

B.2.14.10. Ticket::Frontend::AccountTime

Description	Value
Description:	Activates time accounting.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AccountTime'} = '1';</code>

B.2.14.11. Ticket::Frontend::TimeUnits

Description	Value
Description:	Sets the preferred time units (e.g. work units, hours, minutes).
Group:	Ticket
SubGroup:	Frontend::Agent

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::TimeUnits'} = ' (work units)';</code>

B.2.14.12. Ticket::Frontend::NeedAccountedTime

Description	Value
Description:	Defines if time accounting is mandatory in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NeedAccountedTime'} = '0';</code>

B.2.14.13. Ticket::Frontend::NeedSpellCheck

Description	Value
Description:	Defines if composed messages have to be spell checked in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NeedSpellCheck'} = '0';</code>

B.2.14.14. Ticket::Frontend::NewOwnerSelection

Description	Value
Description:	Shows an owner selection in phone and email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NewOwnerSelection'} = '1';</code>

B.2.14.15. Ticket::Frontend::NewResponsibleSelection

Description	Value
Description:	Show a responsible selection in phone and email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NewResponsibleSelection'} = '1';</code>

B.2.14.16. Ticket::Frontend::NewQueueSelectionType

Description	Value
Description:	Defines the receipt target of the phone ticket and the sender of the email ticket ("Queue" shows all queues, "SystemAddress" displays all system addresses) in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NewQueueSelectionType'} = 'Queue';</code>

B.2.14.17. Ticket::Frontend::NewQueueSelectionString

Description	Value
Description:	Determines the strings that will be shown as receipt (To:) of the phone ticket and as sender (From:) of the email ticket in the agent interface. For Queue as NewQueueSelectionType "<Queue>" shows the names of the queues and for SystemAddress "<Realname> <<Email>>" shows the name and email of the receipt.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NewQueueSelectionString'} = '<Queue>';</code>

B.2.14.18. Ticket::Frontend::NewQueueOwnSelection

Description	Value
Description:	Determines which options will be valid of the receipient (phone ticket) and the sender (email ticket) in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NewQueueOwnSelection'} = { '1' => 'First Queue!', '2' => 'Second Queue!'};</code>

B.2.14.19. Ticket::Frontend::ShowCustomerTickets

Description	Value
Description:	Shows customer history tickets in AgentTicketPhone, AgentTicketEmail and AgentTicketCustomer.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::ShowCustomerTickets'} = '1';</code>

B.2.14.20. CustomerDBLink

Description	Value
Description:	Defines an external link to the database of the customer (e.g. 'http://yourhost/customer.php?CID=\$Data{"CustomerID"}' or ").
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'CustomerDBLink'} = '\$Env{"CGIHandle"}?Action=AgentTicketCusto</code>

B.2.14.21. CustomerDBLinkTarget

Description	Value
Description:	Defines the target attribute in the link to external customer database. E.g. 'target="cdb"'.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerDBLinkTarget'} = "";</code>

B.2.14.22. Frontend::CommonObject###QueueObject

Description	Value
Description:	Path of the file that stores all the settings for the QueueObject object for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::CommonObject'}->{'QueueObject'} = 'Kernel::System::</code>

B.2.14.23. Frontend::CommonObject###TicketObject

Description	Value
Description:	Path of the file that stores all the settings for the TicketObject for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Frontend::CommonObject'}->{'TicketObject'} = 'Kernel::System:'</code>

B.2.14.24. Frontend::CommonParam###Action

Description	Value
Description:	Defines the default used Frontend-Module if no Action parameter given in the url on the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::CommonParam'}->{'Action'} = 'AgentDashboard';</code>

B.2.14.25. Frontend::CommonParam###QueueID

Description	Value
Description:	Default queue ID used by the system in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::CommonParam'}->{'QueueID'} = '0';</code>

B.2.14.26. Frontend::CommonParam###TicketID

Description	Value
Description:	Default ticket ID used by the system in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::CommonParam'}->{'TicketID'} = '';</code>

B.2.15. Frontend::Agent::CustomerSearch

B.2.15.1. Ticket::Frontend::CustomerSearchAutoComplete###Active

Description	Value
Description:	Enables or disables the autocomplete feature for the customer search in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::CustomerSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerSearchAutoComplete'}->{'Active'} =</code>

B.2.15.2.

Ticket::Frontend::CustomerSearchAutoComplete###MinQueryLength

Description	Value
Description:	Sets the minimum number of characters before autocomplete query is sent.
Group:	Ticket
SubGroup:	Frontend::Agent::CustomerSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerSearchAutoComplete'}->{'MinQueryLeng</code>

B.2.15.3. Ticket::Frontend::CustomerSearchAutoComplete###QueryDelay

Description	Value
Description:	Delay time between autocomplete queries.
Group:	Ticket
SubGroup:	Frontend::Agent::CustomerSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerSearchAutoComplete'}->{'QueryDelay'}</code>

B.2.15.4. Ticket::Frontend::CustomerSearchAutoComplete###TypeAhead

Description	Value
Description:	Activates TypeAhead for the autocomplete feature, that enables users to type in whatever speed they desire, without losing any information. Often this means that keystrokes entered will not be displayed on the screen immediately.
Group:	Ticket
SubGroup:	Frontend::Agent::CustomerSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerSearchAutoComplete'}->{'TypeAhead'}</code>

B.2.15.5.**Ticket::Frontend::CustomerSearchAutoComplete###MaxResultsDisplayed**

Description	Value
Description:	Sets the number of search results to be displayed for the autocomplete feature.
Group:	Ticket
SubGroup:	Frontend::Agent::CustomerSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerSearchAutoComplete'}->{'MaxResultsDisplayed'}</code>

B.2.15.6.**Ticket::Frontend::CustomerSearchAutoComplete::DynamicWidth**

Description	Value
Description:	Determines if the search results container for the autocomplete feature should adjust its width dynamically.
Group:	Ticket
SubGroup:	Frontend::Agent::CustomerSearch
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerSearchAutoComplete::DynamicWidth'} =</code>

B.2.16. Frontend::Agent::Dashboard

B.2.16.1. DashboardBackend###0100-TicketPendingReminder

Description	Value
Description:	Parameters for the dashboard backend of the ticket pending reminder overview of the agent interface . "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'DashboardBackend'}->{'0100-TicketPendingReminder'} = { 'Attributes' => 'TicketPendingTimeOlderMinutes=1;StateType=pending remi 'Block' => 'ContentLarge', 'CacheTTLLocal' => '0.5', 'Default' => '1', 'Description' => 'All tickets with a reminder set where the reminder da 'Filter' => 'Locked', 'Group' => "", 'Limit' => '10', 'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric', 'Permission' => 'rw', 'Time' => 'UntilTime', 'Title' => 'Reminder Tickets' };</code>

B.2.16.2. DashboardBackend###0110-TicketEscalation

Description	Value
Description:	Parameters for the dashboard backend of the ticket escalation overview of the agent interface . "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'DashboardBackend'}->{'0110-TicketEscalation'} = { 'Attributes' => 'TicketEscalationTimeOlderMinutes=1;SortBy=EscalationTi 'Block' => 'ContentLarge', 'CacheTTLLocal' => '0.5', 'Default' => '1', 'Description' => 'All escalated tickets', 'Filter' => 'All', 'Group' => "", 'Limit' => '10', 'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric', 'Permission' => 'rw', 'Time' => 'EscalationTime', 'Title' => 'Escalated Tickets' };</pre>

B.2.16.3. DashboardBackend###0120-TicketNew

Description	Value
Description:	Parameters for the dashboard backend of the new tickets overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'DashboardBackend'}->{'0120-TicketNew'} = { 'Attributes' => 'StateType=new;', 'Block' => 'ContentLarge', 'CacheTTLLocal' => '0.5', 'Default' => '1', 'Description' => 'All new tickets, these tickets have not been worked o 'Filter' => 'All', 'Group' => "", 'Limit' => '10', 'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric', 'Permission' => 'rw', 'Time' => 'Age', 'Title' => 'New Tickets' };</pre>

B.2.16.4. DashboardBackend###0130-TicketOpen

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

Description	Value
Description:	Parameters for the dashboard backend of the ticket pending reminder overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'DashboardBackend'}->{'0130-TicketOpen'} = { 'Attributes' => 'StateType=open;', 'Block' => 'ContentLarge', 'CacheTTLLocal' => '0.5', 'Default' => '1', 'Description' => 'All open tickets, these tickets have already been wor 'Filter' => 'All', 'Group' => "", 'Limit' => '10', 'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric', 'Permission' => 'rw', 'Time' => 'Age', 'Title' => 'Open Tickets / Need to be answered' }; </pre>

B.2.16.5. DashboardBackend###0250-TicketStats

Description	Value
Description:	Parameters for the dashboard backend of the ticket stats of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'DashboardBackend'}->{'0250-TicketStats'} = { 'Block' => 'ContentSmall', 'CacheTTL' => '30', 'Closed' => '1', 'Created' => '1', 'Default' => '1', 'Group' => "", 'Module' => 'Kernel::Output::HTML::DashboardTicketStatsGeneric', 'Permission' => 'rw', 'Title' => '7 Day Stats' }; </pre>

B.2.16.6. DashboardBackend###0260-TicketCalendar

Description	Value
Description:	Parameters for the dashboard backend of the ticket calendar of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'DashboardBackend'}->{'0260-TicketCalendar'} = { 'Block' => 'ContentSmall', 'CacheTTL' => '2', 'Default' => '1', 'Group' => "", 'Limit' => '6', 'Module' => 'Kernel::Output::HTML::DashboardCalendar', 'OwnerOnly' => "", 'Permission' => 'rw', 'Title' => 'Upcoming Events' }; </pre>

B.2.17. Frontend::Agent::ModuleMetaHead**B.2.17.1. Frontend::HeaderMetaModule###2-TicketSearch**

Description	Value
Description:	Module to generate html OpenSearch profile for short ticket search in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleMetaHead
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::HeaderMetaModule'}->{'2-TicketSearch'} = { 'Action' => 'AgentTicketSearch', 'Module' => 'Kernel::Output::HTML::HeaderMetaTicketSearch' }; </pre>

B.2.18. Frontend::Agent::ModuleNotify

B.2.18.1. Frontend::NotifyModule###5-Ticket::TicketEscalation

Description	Value
Description:	Module to show notifications and escalations (ShownMax: max. shown escalations, EscalationInMinutes: Show ticket which will escalation in, CacheTime: Cache of calculated escalations in seconds).
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::NotifyModule'}->{'5-Ticket::TicketEscalation'} = { 'CacheTime' => '40', 'EscalationInMinutes' => '120', 'Module' => 'Kernel::Output::HTML::NotificationAgentTicketEscalation', 'ShownMax' => '25' }; </pre>

B.2.19. Frontend::Agent::ModuleRegistration

B.2.19.1. Frontend::Module###AgentTicketQueue

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

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketQueue'} = { 'Description' => 'Overview of all open Tickets', 'Loader' => { 'CSS' => ['Core.AgentTicketQueue.css'] }, 'NavBar' => [{ 'AccessKey' => 'o', 'Block' => "", 'Description' => 'Overview of all open Tickets', 'Link' => 'Action=AgentTicketQueue', 'LinkOption' => "", 'Name' => 'Queue view', 'NavBar' => 'Ticket', 'Prio' => '100', 'Type' => " }, { 'AccessKey' => 't', 'Block' => 'ItemArea', 'Description' => "", 'Link' => 'Action=AgentTicketQueue', 'LinkOption' => "", 'Name' => 'Tickets', 'NavBar' => 'Ticket', 'Prio' => '200', 'Type' => 'Menu' }], 'NavBarName' => 'Ticket', 'Title' => 'QueueView' }; </pre>

B.2.19.2. Frontend::Module###AgentTicketPhone

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketPhone'} = { 'Description' => 'Create new phone picket', 'Loader' => { 'JavaScript' => ['Core.Agent.CustomerSearch.js', 'Core.Agent.TicketAction.js'] }, 'NavBar' => [{ 'AccessKey' => 'n', 'Block' => "", 'Description' => 'Create new phone ticket (inbound)', 'Link' => 'Action=AgentTicketPhone', 'LinkOption' => "", 'Name' => 'New phone ticket', 'NavBar' => 'Ticket', 'Prio' => '200', 'Type' => " }], 'NavBarName' => 'Ticket', 'Title' => 'New phone ticket' }; </pre>

B.2.19.3. Frontend::Module###AgentTicketPhoneOutbound

Description	Value
Description:	Frontend module registration for the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketPhoneOutbound'} = { 'Description' => 'Phone Call', 'Loader' => { 'JavaScript' => ['Core.Agent.TicketAction.js'], 'NavBarName' => 'Ticket', 'Title' => 'Phone-Ticket' }; </pre>

B.2.19.4. Frontend::Module###AgentTicketEmail

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketEmail'} = { 'Description' => 'Create new email ticket', 'Loader' => { 'JavaScript' => ['Core.Agent.CustomerSearch.js', 'Core.Agent.TicketAction.js'] }, 'NavBar' => [{ 'AccessKey' => 'e', 'Block' => "", 'Description' => 'Create new email ticket and send this out (outbound)', 'Link' => 'Action=AgentTicketEmail', 'LinkOption' => "", 'Name' => 'New email ticket', 'NavBar' => 'Ticket', 'Prio' => '210', 'Type' => "" }], 'NavBarName' => 'Ticket', 'Title' => 'New email ticket' }; </pre>

B.2.19.5. Frontend::Module###AgentTicketSearch

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

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketSearch'} = { 'Description' => 'Search Ticket', 'NavBar' => [{ 'AccessKey' => 's', 'Block' => "", 'Description' => 'Search Ticket', 'Link' => 'Action=AgentTicketSearch', 'LinkOption' => 'onclick="window.setTimeout(function(){Core.Agent.Search 'Name' => 'Search', 'NavBar' => 'Ticket', 'Prio' => '300', 'Type' => " }], 'NavBarName' => 'Ticket', 'Title' => 'Search' }; </pre>

B.2.19.6. Frontend::Module###AgentTicketMailbox

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketMailbox'} = { 'Description' => 'compat module for AgentTicketMailbox to AgentTicketLo 'NavBarName' => 'Ticket', 'Title' => " }; </pre>

B.2.19.7. Frontend::Module###AgentTicketLockedView

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketLockedView'} = { 'Description' => 'Locked Tickets', 'NavBarName' => 'Ticket', 'Title' => 'Locked Tickets' }; </pre>

B.2.19.8. Frontend::Module###AgentTicketResponsibleView

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketResponsibleView'} = { 'Description' => 'Responsible Tickets', 'NavBarName' => 'Ticket', 'Title' => 'Responsible Tickets' }; </pre>

B.2.19.9. Frontend::Module###AgentTicketWatchView

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketWatchView'} = { 'Description' => 'Watched Tickets', 'NavBarName' => 'Ticket', 'Title' => 'Watched Tickets' }; </pre>

B.2.19.10. Frontend::Module###AgentCustomerSearch

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

B.2.19.11. Frontend::Module###AgentTicketStatusView

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketStatusView'} = { 'Description' => 'Overview of all open tickets', 'NavBar' => [{ 'AccessKey' => 'v', 'Block' => "", 'Description' => 'Overview of all open Tickets.', 'Link' => 'Action=AgentTicketStatusView', 'LinkOption' => "", 'Name' => 'Status view', 'NavBar' => 'Ticket', 'Prio' => '110', 'Type' => " }], 'NavBarName' => 'Ticket', 'Title' => 'Status view' }; </pre>

B.2.19.12. Frontend::Module###AgentTicketEscalationView

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketEscalationView'} = { 'Description' => 'Overview of all escalated tickets', 'NavBar' => [{ 'AccessKey' => 'e', 'Block' => "", 'Description' => 'Overview Escalated Tickets', 'Link' => 'Action=AgentTicketEscalationView', 'LinkOption' => "", 'Name' => 'Escalation view', 'NavBar' => 'Ticket', 'Prio' => '120', 'Type' => " }], 'NavBarName' => 'Ticket', 'Title' => 'Escalation view' }; </pre>

B.2.19.13. Frontend::Module###AgentZoom

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentZoom'} = { 'Description' => 'compat module for AgentZoom to AgentTicketZoom', 'NavBarName' => 'Ticket', 'Title' => " };</pre>

B.2.19.14. Frontend::Module###AgentTicketZoom

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketZoom'} = { 'Description' => 'Ticket Zoom', 'Loader' => { 'JavaScript' => ['thirdparty/jquery-tablesorter-2.0.5/jquery.tablesorter.js', 'Core.UI.Table.Sort.js', 'Core.Agent.TicketZoom.js'] }, 'NavBarName' => 'Ticket', 'Title' => 'Zoom' };</pre>

B.2.19.15. Frontend::Module###AgentTicketAttachment

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketAttachment'} = { 'Description' => 'To download attachments', 'NavBarName' => 'Ticket', 'Title' => " };</pre>

B.2.19.16. Frontend::Module###AgentTicketPlain

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketPlain'} = { 'Description' => 'Ticket plain view of an email', 'NavBarName' => 'Ticket', 'Title' => 'Plain' }; </pre>

B.2.19.17. Frontend::Module###AgentTicketNote

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketNote'} = { 'Description' => 'Ticket Note', 'Loader' => { 'JavaScript' => ['Core.Agent.TicketAction.js'] }, 'NavBarName' => 'Ticket', 'Title' => 'Note' }; </pre>

B.2.19.18. Frontend::Module###AgentTicketMerge

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketMerge'} = { 'Description' => 'Ticket Merge', 'NavBarName' => 'Ticket', 'Title' => 'Merge' }; </pre>

B.2.19.19. Frontend::Module###AgentTicketPending

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketPending'} = { 'Description' => 'Ticket Pending', 'Loader' => { 'JavaScript' => ['Core.Agent.TicketAction.js'] }, 'NavBarName' => 'Ticket', 'Title' => 'Pending' };</pre>

B.2.19.20. Frontend::Module###AgentTicketWatcher

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketWatcher'} = { 'Description' => 'A TicketWatcher Module', 'NavBarName' => 'Ticket-Watcher', 'Title' => 'Ticket-Watcher' }; </pre>

B.2.19.21. Frontend::Module###AgentTicketPriority

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

Description	Value
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketPriority'} = { 'Description' => 'Ticket Priority', 'Loader' => { 'JavaScript' => ['Core.Agent.TicketAction.js'] }, 'NavBarName' => 'Ticket', 'Title' => 'Priority' }; </pre>

B.2.19.22. Frontend::Module###AgentTicketLock

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketLock'} = { 'Description' => 'Ticket Lock', 'NavBarName' => 'Ticket', 'Title' => 'Lock' }; </pre>

B.2.19.23. Frontend::Module###AgentTicketMove

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketMove'} = { 'Description' => 'Ticket Move', 'Loader' => { 'JavaScript' => ['Core.Agent.TicketAction.js'] }, 'NavBarName' => 'Ticket', 'Title' => 'Move' }; </pre>

B.2.19.24. Frontend::Module###AgentTicketHistory

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketHistory'} = { 'Description' => 'Ticket History', 'NavBarName' => 'Ticket', 'Title' => 'History' }; </pre>

B.2.19.25. Frontend::Module###AgentTicketOwner

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketOwner'} = { 'Description' => 'Ticket Owner', 'Loader' => { 'JavaScript' => ['Core.Agent.TicketAction.js'] }, 'NavBarName' => 'Ticket', 'Title' => 'Owner' }; </pre>

B.2.19.26. Frontend::Module###AgentTicketResponsible

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketResponsible'} = { 'Description' => 'Ticket Responsible', 'Loader' => { 'JavaScript' => ['Core.Agent.TicketAction.js'] }, 'NavBarName' => 'Ticket', 'Title' => 'Responsible' }; </pre>

B.2.19.27. Frontend::Module###AgentTicketCompose

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketCompose'} = { 'Description' => 'Ticket Compose email Answer', 'Loader' => { 'JavaScript' => ['Core.Agent.CustomerSearch.js', 'Core.Agent.TicketAction.js'] }, 'NavBarName' => 'Ticket', 'Title' => 'Compose' }; </pre>

B.2.19.28. Frontend::Module###AgentTicketBounce

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketBounce'} = { 'Description' => 'Ticket Compose Bounce Email', 'NavBarName' => 'Ticket', 'Title' => 'Bounce' }; </pre>

B.2.19.29. Frontend::Module###AgentTicketForward

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketForward'} = { 'Description' => 'Ticket Forward Email', 'Loader' => { 'JavaScript' => ['Core.Agent.CustomerSearch.js', 'Core.Agent.TicketAction.js'] }, 'NavBarName' => 'Ticket', 'Title' => 'Forward' }; </pre>

B.2.19.30. Frontend::Module###AgentTicketCustomer

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketCustomer'} = { 'Description' => 'Ticket Customer', 'Loader' => { 'JavaScript' => ['Core.Agent.CustomerSearch.js', 'Core.Agent.TicketAction.js'] }, 'NavBarName' => 'Ticket', 'Title' => 'Customer' }; </pre>

B.2.19.31. Frontend::Module###AgentTicketClose

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketClose'} = { 'Description' => 'Ticket Close', 'Loader' => { 'JavaScript' => ['Core.Agent.TicketAction.js'] }, 'NavBarName' => 'Ticket', 'Title' => 'Close' }; </pre>

B.2.19.32. Frontend::Module###AgentTicketFreeText

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

Description	Value
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketFreeText'} = { 'Description' => 'Ticket FreeText', 'Loader' => { 'JavaScript' => ['Core.Agent.TicketAction.js'] }, 'NavBarName' => 'Ticket', 'Title' => 'Free Fields' }; </pre>

B.2.19.33. Frontend::Module###AgentTicketPrint

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketPrint'} = { 'Description' => 'Ticket Print', 'NavBarName' => 'Ticket', 'Title' => 'Print' }; </pre>

B.2.19.34. Frontend::Module###AgentTicketBulk

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketBulk'} = { 'Description' => 'Ticket bulk module', 'Loader' => { 'JavaScript' => ['Core.Agent.TicketAction.js'] }, 'NavBarName' => 'Ticket', 'Title' => 'Bulk-Action' }; </pre>

B.2.20. Frontend::Agent::Preferences

B.2.20.1. PreferencesGroups###NewTicketNotify

Description	Value
Description:	Parameters for the NewTicketNotify object in the preferences view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'NewTicketNotify'} = { 'Active' => '1', 'Column' => 'Email Settings', 'Data' => { '0' => 'No', '1' => 'Yes' }, 'DataSelected' => '0', 'Desc' => 'Send me a notification if there is a new ticket in "My Queue"', 'Key' => 'Send new ticket notifications', 'Label' => 'New ticket notification', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserSendNewTicketNotification', 'Prio' => '1000' }; </pre>

B.2.20.2. PreferencesGroups###FollowUpNotify

Description	Value
Description:	Parameters for the FollowUpNotify object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'FollowUpNotify'} = { 'Active' => '1', 'Column' => 'Email Settings', 'Data' => { '0' => 'No', '1' => 'Yes' }, 'DataSelected' => '0', 'Desc' => 'Send me a notification if a customer sends a follow up and I am available', 'Key' => 'Send ticket follow up notifications', 'Label' => 'Ticket follow up notification', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserSendFollowUpNotification', 'Prio' => '2000' }; </pre>

B.2.20.3. PreferencesGroups###LockTimeoutNotify

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

Description	Value
Description:	Parameters for the LockTimeoutNotify object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'LockTimeoutNotify'} = { 'Active' => '1', 'Column' => 'Email Settings', 'Data' => { '0' => 'No', '1' => 'Yes' }, 'DataSelected' => '0', 'Desc' => 'Send me a notification if a ticket is unlocked by the system', 'Key' => 'Send ticket lock timeout notifications', 'Label' => 'Ticket lock timeout notification', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserSendLockTimeoutNotification', 'Prio' => '3000' }; </pre>

B.2.20.4. PreferencesGroups###MoveNotify

Description	Value
Description:	Parameters for the MoveNotify object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'MoveNotify'} = { 'Active' => '1', 'Column' => 'Email Settings', 'Data' => { '0' => 'No', '1' => 'Yes' }, 'DataSelected' => '0', 'Desc' => 'Send me a notification if a ticket is moved into one of "My', 'Key' => 'Send ticket move notifications', 'Label' => 'Ticket move notification', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserSendMoveNotification', 'Prio' => '4000' }; </pre>

B.2.20.5. PreferencesGroups###WatcherNotify

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

Description	Value
Description:	Parameters for the WatcherNotify object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'WatcherNotify'} = { 'Active' => '1', 'Column' => 'Email Settings', 'Data' => { '0' => 'No', '1' => 'Yes' }, 'DataSelected' => '0', 'Desc' => 'Send me the same notifications for my watched tickets that t 'Key' => 'Send ticket watch notifications', 'Label' => 'Ticket watch notification', 'Module' => 'Kernel::Output::HTML::PreferencesTicketWatcher', 'PrefKey' => 'UserSendWatcherNotification', 'Prio' => '5000' }; </pre>

B.2.20.6. PreferencesGroups###CustomQueue

Description	Value
Description:	Parameters for the CustomQueue object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'CustomQueue'} = { 'Active' => '1', 'Column' => 'Other Settings', 'Desc' => 'Your queue selection of your favourite queues. You also get 'Key' => 'My Queues', 'Label' => 'My Queues', 'Module' => 'Kernel::Output::HTML::PreferencesCustomQueue', 'Permission' => 'ro', 'Prio' => '1000' }; </pre>

B.2.20.7. PreferencesGroups###RefreshTime

Description	Value
Description:	Parameters for the RefreshTime object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences

Description	Value
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'RefreshTime'} = { 'Active' => '1', 'Column' => 'Other Settings', 'Data' => { '0' => 'off', '10' => '10 minutes', '15' => '15 minutes', '2' => ' 2 minutes', '5' => ' 5 minutes', '7' => ' 7 minutes' }, 'DataSelected' => '0', 'Desc' => 'If enabled, the QueueView will automatically refresh after t 'Key' => 'Refresh QueueView after', 'Label' => 'QueueView Refresh Time', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserRefreshTime', 'Prio' => '2000' }; </pre>

B.2.20.8. PreferencesGroups###TicketOverviewSmallPageShown

Description	Value
Description:	Parameters for the pages (in which the tickets are shown) of the small ticket overview.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'TicketOverviewSmallPageShown'} = { 'Active' => '0', 'Column' => 'Other Settings', 'Data' => { '10' => '10', '15' => '15', '20' => '20', '25' => '25', '30' => '30', '35' => '35' }, 'DataSelected' => '25', 'Key' => 'Ticket limit per page for Ticket Overview "Small"', 'Label' => 'Ticket Overview "Small" Limit', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserTicketOverviewSmallPageShown', 'Prio' => '8000' }; </pre>

B.2.20.9. PreferencesGroups###TicketOverviewMediumPageShown

Description	Value
Description:	Parameters for the pages (in which the tickets are shown) of the medium ticket overview.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'TicketOverviewMediumPageShown'} = { 'Active' => '0', 'Column' => 'Other Settings', 'Data' => { '10' => '10', '15' => '15', '20' => '20', '25' => '25', '30' => '30', '35' => '35' }, 'DataSelected' => '20', 'Key' => 'Ticket limit per page for Ticket Overview "Medium"', 'Label' => 'Ticket Overview "Medium" Limit', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserTicketOverviewMediumPageShown', 'Prio' => '8100' }; </pre>

B.2.20.10. PreferencesGroups###TicketOverviewPreviewPageShown

Description	Value
Description:	Parameters for the pages (in which the tickets are shown) of the ticket preview overview.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'TicketOverviewPreviewPageShown'} = { 'Active' => '0', 'Column' => 'Other Settings', 'Data' => { '10' => '10', '15' => '15', '20' => '20', '25' => '25', '30' => '30', '35' => '35' }, 'DataSelected' => '15', 'Key' => 'Ticket limit per page for Ticket Overview "Preview"', 'Label' => 'Ticket Overview "Preview" Limit', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserTicketOverviewPreviewPageShown', 'Prio' => '8200' }; </pre>

B.2.20.11. PreferencesGroups###CreateNextMask

Description	Value
Description:	Parameters for the CreateNextMask object in the preference view of the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'CreateNextMask'} = { 'Active' => '1', 'Column' => 'Other Settings', 'Data' => { " => 'CreateTicket', 'AgentTicketZoom' => 'TicketZoom' }, 'DataSelected' => "", 'Key' => 'Show this screen after I created a new ticket', 'Label' => 'Screen after new ticket', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserCreateNextMask', 'Prio' => '3000' };</pre>

B.2.21. Frontend::Agent::SearchRouter

B.2.21.1. Frontend::Search###Ticket

Description	Value
Description:	Search backend router.
Group:	Ticket
SubGroup:	Frontend::Agent::SearchRouter
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Search'}->{'Ticket'} = { '^AgentTicket' => 'Action=AgentTicketSearch;Subaction=AJAX' };</pre>

B.2.22. Frontend::Agent::Ticket::ArticleAttachmentModule

B.2.22.1. Ticket::Frontend::ArticleAttachmentModule###1-Download

Description	Value
Description:	Shows a link to download article attachments in the zoom view of the article in the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ArticleAttachmentModule
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ArticleAttachmentModule'}->{'1-Download'} = 'Module' => 'Kernel::Output::HTML::ArticleAttachmentDownload' };</pre>

B.2.22.2. Ticket::Frontend::ArticleAttachmentModule###2-HTML-Viewer

Description	Value
Description:	Shows a link to access article attachments via a html online viewer in the zoom view of the article in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleAttachmentModule
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ArticleAttachmentModule'}->{'2-HTML-Viewer'} 'Module' => 'Kernel::Output::HTML::ArticleAttachmentHTMLViewer' };</pre>

B.2.23. Frontend::Agent::Ticket::ArticleComposeModule

B.2.23.1. Ticket::Frontend::ArticleComposeModule###1-SignEmail

Description	Value
Description:	Module to compose signed messages (PGP or S/MIME).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleComposeModule
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ArticleComposeModule'}->{'1-SignEmail'} = { 'Module' => 'Kernel::Output::HTML::ArticleComposeSign' }; };</pre>

B.2.23.2. Ticket::Frontend::ArticleComposeModule###2-CryptEmail

Description	Value
Description:	Module to crypt composed messages (PGP or S/MIME).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleComposeModule
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ArticleComposeModule'}->{'2-CryptEmail'} = 'Module' => 'Kernel::Output::HTML::ArticleComposeCrypt' };</pre>

B.2.24. Frontend::Agent::Ticket::ArticleViewModule**B.2.24.1. 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
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ArticleViewModule'}->{'1-PGP'} = { 'Module' => 'Kernel::Output::HTML::ArticleCheckPGP' }; </pre>

B.2.24.2. Ticket::Frontend::ArticleViewModule###1-SMIME

Description	Value
Description:	Agent interface module to check incoming emails in the Ticket-Zoom-View if the S/MIME-key is available and true.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModule
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ArticleViewModule'}->{'1-SMIME'} = { 'Module' => 'Kernel::Output::HTML::ArticleCheckSMIME' };</pre>

B.2.25. Frontend::Agent::Ticket::ArticleViewModulePre

B.2.25.1. Ticket::Frontend::ArticlePreViewModule###1-PGP

Description	Value
Description:	Agent interface article notification module to check PGP.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModulePre
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ArticlePreViewModule'}->{'1-PGP'} = { 'Module' => 'Kernel::Output::HTML::ArticleCheckPGP' };</pre>

B.2.25.2. Ticket::Frontend::ArticlePreViewModule###1-SMIME

Description	Value
Description:	Agent interface article notification module to check S/MIME.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModulePre
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ArticlePreViewModule'}->{'1-SMIME'} = { 'Module' => 'Kernel::Output::HTML::ArticleCheckSMIME' };</pre>

B.2.26. Frontend::Agent::Ticket::MenuModule

B.2.26.1. Ticket::Frontend::MenuModule###000-Back

Description	Value
Description:	Shows a link in the menu to go back in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'000-Back'} = { 'Action' => "", 'Description' => 'Back', 'Link' => '\$Env{"LastScreenOverview"};TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Back', 'PopupType' => "", 'Target' => "" }; </pre>

B.2.26.2. Ticket::Frontend::MenuModule###100-Lock

Description	Value
Description:	Shows a link in the menu to lock/unlock tickets in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'100-Lock'} = { 'Action' => 'AgentTicketLock', 'Module' => 'Kernel::Output::HTML::TicketMenuLock', 'Name' => 'Lock', 'Target' => "" }; </pre>

B.2.26.3. Ticket::Frontend::MenuModule###200-History

Description	Value
Description:	Shows a link in the menu to access the history of a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'200-History'} = { 'Action' => 'AgentTicketHistory', 'Description' => 'Shows the ticket history!', 'Link' => 'Action=AgentTicketHistory;TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'History', 'PopupType' => 'TicketHistory', 'Target' => " "};</pre>

B.2.26.4. Ticket::Frontend::MenuModule###210-Print

Description	Value
Description:	Shows a link in the menu to print a ticket or an article in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'210-Print'} = { 'Action' => 'AgentTicketPrint', 'Description' => 'Print this ticket!', 'Link' => 'Action=AgentTicketPrint;TicketID=\$QData{"TicketID"}', 'LinkParam' => 'target="print"', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Print', 'PopupType' => 'TicketAction', 'Target' => " "};</pre>

B.2.26.5. Ticket::Frontend::MenuModule###300-Priority

Description	Value
Description:	Shows a link in the menu to see the priority of a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'300-Priority'} = { 'Action' => 'AgentTicketPriority', 'Description' => 'Change the ticket priority!', 'Link' => 'Action=AgentTicketPriority;TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Priority', 'PopupType' => 'TicketAction', 'Target' => " "};</pre>

B.2.26.6. Ticket::Frontend::MenuModule###310-FreeText

Description	Value
Description:	Shows a link in the menu to add a free text field in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'310-FreeText'} = { 'Action' => 'AgentTicketFreeText', 'Description' => 'Change the ticket free fields!', 'Link' => 'Action=AgentTicketFreeText;TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Free Fields', 'PopupType' => 'TicketAction', 'Target' => " "};</pre>

B.2.26.7. Ticket::Frontend::MenuModule###320-Link

Description	Value
Description:	Shows a link in the menu that allows linking a ticket with another object in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'320-Link'} = { 'Action' => 'AgentLinkObject', 'Description' => 'Link this ticket to other objects!', 'Link' => 'Action=AgentLinkObject;SourceObject=Ticket;SourceKey=\$QData{ 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Link', 'PopupType' => 'TicketAction', 'Target' => " " }; </pre>

B.2.26.8. Ticket::Frontend::MenuModule###400-Owner

Description	Value
Description:	Shows a link in the menu to see the owner of a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'400-Owner'} = { 'Action' => 'AgentTicketOwner', 'Description' => 'Change the ticket owner!', 'Link' => 'Action=AgentTicketOwner;TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Owner', 'PopupType' => 'TicketAction', 'Target' => " " }; </pre>

B.2.26.9. Ticket::Frontend::MenuModule###410-Responsible

Description	Value
Description:	Shows a link in the menu to see the responsible agent of a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'410-Responsible'} = { 'Action' => 'AgentTicketResponsible', 'Description' => 'Change the ticket responsible!', 'Link' => 'Action=AgentTicketResponsible;TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuResponsible', 'Name' => 'Responsible', 'PopupType' => 'TicketAction', 'Target' => " "};</pre>

B.2.26.10. Ticket::Frontend::MenuModule###420-Customer

Description	Value
Description:	Shows a link in the menu to see the customer who requested the ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'420-Customer'} = { 'Action' => 'AgentTicketCustomer', 'Description' => 'Change the ticket customer!', 'Link' => 'Action=AgentTicketCustomer;TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Customer', 'PopupType' => 'TicketAction', 'Target' => " "};</pre>

B.2.26.11. Ticket::Frontend::MenuModule###420-Note

Description	Value
Description:	Shows a link in the menu to add a note in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'420-Note'} = { 'Action' => 'AgentTicketNote', 'Description' => 'Add a note to this ticket!', 'Link' => 'Action=AgentTicketNote;TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Note', 'PopupType' => 'TicketAction', 'Target' => " ";</pre>

B.2.26.12. Ticket::Frontend::MenuModule###430-Merge

Description	Value
Description:	Shows a link in the menu that allows merging tickets in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'430-Merge'} = { 'Action' => 'AgentTicketMerge', 'Description' => 'Merge this ticket!', 'Link' => 'Action=AgentTicketMerge;TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Merge', 'PopupType' => 'TicketAction', 'Target' => " ";</pre>

B.2.26.13. Ticket::Frontend::MenuModule###440-Pending

Description	Value
Description:	Shows a link in the menu to set a ticket as pending in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'440-Pending'} = { 'Action' => 'AgentTicketPending', 'Description' => 'Set this ticket to pending!', 'Link' => 'Action=AgentTicketPending;TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Pending', 'PopupType' => 'TicketAction', 'Target' => " "};</pre>

B.2.26.14. Ticket::Frontend::MenuModule###448-Watch

Description	Value
Description:	Shows a link in the menu for subscribing / unsubscribing from a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'448-Watch'} = { 'Action' => 'AgentTicketWatcher', 'Module' => 'Kernel::Output::HTML::TicketMenuTicketWatcher', 'Name' => 'Watch', 'Target' => " "};</pre>

B.2.26.15. Ticket::Frontend::MenuModule###450-Close

Description	Value
Description:	Shows a link in the menu to close a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'450-Close'} = { 'Action' => 'AgentTicketClose', 'Description' => 'Close this ticket!', 'Link' => 'Action=AgentTicketClose;TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Close', 'PopupType' => 'TicketAction', 'Target' => " "};</pre>

B.2.26.16. Ticket::Frontend::MenuModule###460-Delete

Description	Value
Description:	Shows a link in the menu to delete a ticket in the ticket zoom view of the agent interface. Additional access control to show or not show this link can be done by using Key "Group" and Content like "rw:group1;move_into:group2".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'460-Delete'} = { 'Action' => 'AgentTicketMove', 'Description' => 'Delete this ticket!', 'Link' => 'Action=AgentTicketMove;TicketID=\$Data{"TicketID"};DestQueue= 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Delete', 'PopupType' => "", 'Target' => " };</pre>

B.2.26.17. Ticket::Frontend::MenuModule###470-Spam

Description	Value
Description:	Shows a link to set a ticket as spam in the ticket zoom view of the agent interface. Additional access control to show or not show this link can be done by using Key "Group" and Content like "rw:group1;move_into:group2".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'470-Spam'} = { 'Action' => 'AgentTicketMove', 'Description' => 'Mark as Spam!', 'Link' => 'Action=AgentTicketMove;TicketID=\$Data{"TicketID"};DestQueue= 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Spam', 'PopupType' => "", 'Target' => " };</pre>

B.2.27. Frontend::Agent::Ticket::MenuModulePre

B.2.27.1. Ticket::Frontend::PreMenuModule###100-Lock

Description	Value
Description:	Shows a link in the menu to lock / unlock a ticket in the ticket overviews of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::PreMenuModule'}->{'100-Lock'} = { 'Action' => 'AgentTicketLock', 'Module' => 'Kernel::Output::HTML::TicketMenuLock', 'Name' => 'Lock', 'PopupType' => "", 'Target' => "" }; </pre>

B.2.27.2. Ticket::Frontend::PreMenuModule###200-Zoom

Description	Value
Description:	Shows a link in the menu to zoom a ticket in the ticket overviews of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::PreMenuModule'}->{'200-Zoom'} = { 'Action' => 'AgentTicketZoom', 'Description' => 'Look into a ticket!', 'Link' => 'Action=AgentTicketZoom;TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Zoom', 'PopupType' => "", 'Target' => "" }; </pre>

B.2.27.3. Ticket::Frontend::PreMenuModule###210-History

Description	Value
Description:	Shows a link in the menu to see the history of a ticket in every ticket overview of the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::PreMenuModule'}->{'210-History'} = { 'Action' => 'AgentTicketHistory', 'Description' => 'Shows the ticket history!', 'Link' => 'Action=AgentTicketHistory;TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'History', 'PopupType' => 'TicketHistory', 'Target' => " "};</pre>

B.2.27.4. Ticket::Frontend::PreMenuModule###300-Priority

Description	Value
Description:	Shows a link in the menu to set the priority of a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::PreMenuModule'}->{'300-Priority'} = { 'Action' => 'AgentTicketPriority', 'Description' => 'Change the ticket priority!', 'Link' => 'Action=AgentTicketPriority;TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Priority', 'PopupType' => 'TicketAction', 'Target' => " "};</pre>

B.2.27.5. Ticket::Frontend::PreMenuModule###420-Note

Description	Value
Description:	Shows a link in the menu to add a note to a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::PreMenuModule'}->{'420-Note'} = { 'Action' => 'AgentTicketNote', 'Description' => 'Add a note to this ticket!', 'Link' => 'Action=AgentTicketNote;TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Note', 'PopupType' => 'TicketAction', 'Target' => " "};</pre>

B.2.27.6. Ticket::Frontend::PreMenuModule###440-Close

Description	Value
Description:	Shows a link in the menu to close a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::PreMenuModule'}->{'440-Close'} = { 'Action' => 'AgentTicketClose', 'Description' => 'Close this ticket!', 'Link' => 'Action=AgentTicketClose;TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Close', 'PopupType' => 'TicketAction', 'Target' => " "};</pre>

B.2.27.7. Ticket::Frontend::PreMenuModule###445-Move

Description	Value
Description:	Shows a link in the menu to move a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::PreMenuModule'}->{'445-Move'} = { 'Action' => 'AgentTicketMove', 'Description' => 'Change queue!', 'Module' => 'Kernel::Output::HTML::TicketMenuMove', 'Name' => 'Move' }; </pre>

B.2.27.8. Ticket::Frontend::PreMenuModule###450-Delete

Description	Value
Description:	Shows a link in the menu to delete a ticket in every ticket overview of the agent interface. Additional access control to show or not show this link can be done by using Key "Group" and Content like "rw:group1;move_into:group2".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::PreMenuModule'}->{'450-Delete'} = { 'Action' => 'AgentTicketMove', 'Description' => 'Delete this ticket!', 'Link' => 'Action=AgentTicketMove;TicketID=\$Data{"TicketID"};DestQueue= 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Delete', 'PopupType' => "", 'Target' => "" }; </pre>

B.2.27.9. Ticket::Frontend::PreMenuModule###460-Spam

Description	Value
Description:	Shows a link in the menu to set a ticket as spam in every ticket overview of the agent interface. Additional access control to show or not show this link can be done by using Key "Group" and Content like "rw:group1;move_into:group2".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::PreMenuModule'}->{'460-Spam'} = { 'Action' => 'AgentTicketMove', 'Description' => 'Mark as Spam!', 'Link' => 'Action=AgentTicketMove;TicketID=\$Data{"TicketID"};DestQueue= 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Spam', 'PopupType' => "", 'Target' => "" };</pre>

B.2.28. Frontend::Agent::Ticket::ViewBounce

B.2.28.1. Ticket::Frontend::AgentTicketBounce###Permission

Description	Value
Description:	Required permissions to use the ticket bounce screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'Permission'} = 'boun</pre>

B.2.28.2. Ticket::Frontend::AgentTicketBounce###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket bounce screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'RequiredLock'} = '1'</pre>

B.2.28.3. Ticket::Frontend::AgentTicketBounce###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after being bounced, in the ticket bounce screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'StateDefault'} = 'cl</code>

B.2.28.4. Ticket::Frontend::AgentTicketBounce###StateType

Description	Value
Description:	Defines the next state of a ticket after being bounced, in the ticket bounce screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'StateType'} = ['open', 'closed'];</code>

B.2.28.5. Ticket::Frontend::BounceText

Description	Value
Description:	Defines the default ticket bounced notification for customer/sender in the ticket bounce screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::BounceText'} = 'Your email with ticket numb</code>

B.2.29. Frontend::Agent::Ticket::ViewBulk

B.2.29.1. Ticket::Frontend::AgentTicketBulk###RequiredLock

Description	Value
Description:	Automatically lock and set owner to current Agent after selecting for an Bulk Action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'RequiredLock'} = '1';</code>

B.2.29.2. Ticket::Frontend::AgentTicketBulk###Owner

Description	Value
Description:	Sets the ticket owner in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Owner'} = '1';</code>

B.2.29.3. Ticket::Frontend::AgentTicketBulk###Responsible

Description	Value
Description:	Sets the responsible agent of the ticket in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Responsible'} = '1';</code>

B.2.29.4. Ticket::Frontend::AgentTicketBulk###State

Description	Value
Description:	If a note is added by an agent, sets the state of a ticket in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'State'} = '1';</code>

B.2.29.5. Ticket::Frontend::AgentTicketBulk###StateType

Description	Value
Description:	Defines the next state of a ticket after adding a note, in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'StateType'} = ['open', 'closed', 'pending reminder', 'pending auto'];</code>

B.2.29.6. Ticket::Frontend::AgentTicketBulk###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after adding a note, in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'StateDefault'} = 'open'</code>

B.2.29.7. Ticket::Frontend::AgentTicketBulk###Priority

Description	Value
Description:	Shows the ticket priority options in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Priority'} = '1';</code>

B.2.29.8. Ticket::Frontend::AgentTicketBulk###PriorityDefault

Description	Value
Description:	Defines the default ticket priority in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'PriorityDefault'} = '3';</code>

B.2.29.9. Ticket::Frontend::AgentTicketBulk###ArticleTypeDefault

Description	Value
Description:	Defines the default type of the note in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'ArticleTypeDefault'} =</code>

B.2.29.10. Ticket::Frontend::AgentTicketBulk###ArticleTypes

Description	Value
Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'ArticleTypes'} = { 'note-external' => '1', 'note-internal' => '1', 'note-report' => '0' }; </pre>

B.2.30. Frontend::Agent::Ticket::ViewClose**B.2.30.1. Ticket::Frontend::AgentTicketClose###Permission**

Description	Value
Description:	Required permissions to use the close ticket screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Permission'} = 'close </pre>

B.2.30.2. Ticket::Frontend::AgentTicketClose###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the close ticket screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'RequiredLock'} = '1';</code>

B.2.30.3. Ticket::Frontend::AgentTicketClose###TicketType

Description	Value
Description:	Sets the ticket type in the close ticket screen of the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'TicketType'} = '0';</code>

B.2.30.4. Ticket::Frontend::AgentTicketClose###Service

Description	Value
Description:	Sets the service in the close ticket screen of the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Service'} = '0';</code>

B.2.30.5. Ticket::Frontend::AgentTicketClose###Owner

Description	Value
Description:	Sets the ticket owner in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Owner'} = '0';</code>

B.2.30.6. Ticket::Frontend::AgentTicketClose###OwnerMandatory

Description	Value
Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'OwnerMandatory'} = '0';</code>

B.2.30.7. Ticket::Frontend::AgentTicketClose###Responsible

Description	Value
Description:	Sets the responsible agent of the ticket in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Responsible'} = '0';</code>

B.2.30.8. Ticket::Frontend::AgentTicketClose###State

Description	Value
Description:	If a note is added by an agent, sets the state of a ticket in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'State'} = '1';</code>

B.2.30.9. Ticket::Frontend::AgentTicketClose###StateType

Description	Value
Description:	Defines the next state of a ticket after adding a note, in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'StateType'} = ['closed'];</code>

B.2.30.10. Ticket::Frontend::AgentTicketClose###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after adding a note, in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'StateDefault'} = 'closed';</code>

B.2.30.11. Ticket::Frontend::AgentTicketClose###Note

Description	Value
Description:	Allows adding notes in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Note'} = '1';</code>

B.2.30.12. Ticket::Frontend::AgentTicketClose###Subject

Description	Value
Description:	Sets the default subject for notes added in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Subject'} = '\$Text{"C</code>

B.2.30.13. Ticket::Frontend::AgentTicketClose###Body

Description	Value
Description:	Sets the default body text for notes added in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Body'} = '';</code>

B.2.30.14. Ticket::Frontend::AgentTicketClose###InvolvedAgent

Description	Value
Description:	Shows a list of all the involved agents on this ticket, in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0

Description	Value
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'InvolvedAgent'} = '0';</code>

B.2.30.15. Ticket::Frontend::AgentTicketClose###InformAgent

Description	Value
Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'InformAgent'} = '0';</code>

B.2.30.16. Ticket::Frontend::AgentTicketClose###ArticleTypeDefault

Description	Value
Description:	Defines the default type of the note in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ArticleTypeDefault'} =</code>

B.2.30.17. Ticket::Frontend::AgentTicketClose###ArticleTypes

Description	Value
Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose

Description	Value
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };</pre>

B.2.30.18. Ticket::Frontend::AgentTicketClose###Priority

Description	Value
Description:	Shows the ticket priority options in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Priority'} = '0';</pre>

B.2.30.19. Ticket::Frontend::AgentTicketClose###PriorityDefault

Description	Value
Description:	Defines the default ticket priority in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'PriorityDefault'} = '0';</pre>

B.2.30.20. Ticket::Frontend::AgentTicketClose###Title

Description	Value
Description:	Shows the title fields in the close ticket screen of the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Title'} = '0';</code>

B.2.30.21. Ticket::Frontend::AgentTicketClose###TicketFreeText

Description	Value
Description:	Ticket free text options shown in the close ticket screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'TicketFreeText'} = { '1' => '0', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0', '15' => '0', '16' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' };</code>

B.2.30.22. Ticket::Frontend::AgentTicketClose###TicketFreeTime

Description	Value
Description:	Ticket free time options shown in the close ticket screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'TicketFreeTime'} = { '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' };</code>

B.2.30.23. Ticket::Frontend::AgentTicketClose###ArticleFreeText

Description	Value
Description:	Article free text options shown in the close ticket screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ArticleFreeText'} = { '1' => '0', '2' => '0', '3' => '0'};</pre>

B.2.30.24. Ticket::Frontend::AgentTicketClose###HistoryType

Description	Value
Description:	Defines the history type for the close ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'HistoryType'} = 'AddN</pre>

B.2.30.25. Ticket::Frontend::AgentTicketClose###HistoryComment

Description	Value
Description:	Defines the history comment for the close ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'HistoryComment'} = '%</pre>

B.2.31. Frontend::Agent::Ticket::ViewCompose

B.2.31.1. Ticket::Frontend::AgentTicketCompose###Permission

Description	Value
Description:	Required permissions to use the ticket compose screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'Permission'} = 'com</code>

B.2.31.2. Ticket::Frontend::AgentTicketCompose###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket compose screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'RequiredLock'} = '1</code>

B.2.31.3. Ticket::Frontend::AgentTicketCompose###StateDefault

Description	Value
Description:	Defines the default next state of a ticket if it is composed / answered in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'StateDefault'} = 'o</code>

B.2.31.4. Ticket::Frontend::AgentTicketCompose###StateType

Description	Value
Description:	Defines the next possible states after composing / answering a ticket in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'StateType'} = ['open', 'closed', 'pending auto', 'pending reminder'];</pre>

B.2.31.5. Ticket::Frontend::ResponseFormat

Description	Value
Description:	Defines the format of responses in the ticket compose screen of the agent interface (\$QData{"OrigFrom"} is From 1:1, \$QData{"OrigFromName"} is only realname of From).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ResponseFormat'} = '\$QData{"Salutation"} \$QData{"StdResponse"} \$QData{"Signature"} \$TimeShort{"\$QData{"Created"}} - \$QData{"OrigFromName"} \$Text{"wrote"} \$QData{"Body"} ' ;</pre>

B.2.31.6. Ticket::Frontend::Quote

Description	Value
Description:	Defines the used character for email quotes in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::Quote'} = '>' ;</pre>

B.2.31.7. Ticket::Frontend::ComposeAddCustomerAddress

Description	Value
Description:	Adds customers email addresses to recipients in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::ComposeAddCustomerAddress'} = '1';</code>

B.2.31.8. Ticket::Frontend::ComposeReplaceSenderAddress

Description	Value
Description:	Replaces the original sender with current customer's email address on compose answer in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::ComposeReplaceSenderAddress'} = '0';</code>

B.2.31.9. Ticket::Frontend::ComposeExcludeCcRecipients

Description	Value
Description:	Uses Cc recipients in reply Cc list on compose an email answer in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::ComposeExcludeCcRecipients'} = '0';</code>

B.2.31.10. Ticket::Frontend::AgentTicketCompose###TicketFreeText

Description	Value
Description:	Ticket free text options shown in the ticket compose screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'TicketFreeText'} = '1' => '0', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0', '15' => '0', '16' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' };</pre>

B.2.31.11. Ticket::Frontend::AgentTicketCompose###TicketFreeTime

Description	Value
Description:	Ticket free time options shown in the ticket compose screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'TicketFreeTime'} = '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

B.2.31.12. Ticket::Frontend::AgentTicketCompose###ArticleFreeText

Description	Value
Description:	Article free text options shown in the ticket compose screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose

Description	Value
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'ArticleFreeText'} = '1' => '0', '2' => '0', '3' => '0' };</code>

B.2.32. Frontend::Agent::Ticket::ViewCustomer

B.2.32.1. Ticket::Frontend::AgentTicketCustomer###Permission

Description	Value
Description:	Required permissions to change the customer of a ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCustomer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketCustomer'}->{'Permission'} = 'cu</code>

B.2.32.2. Ticket::Frontend::AgentTicketCustomer###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required to change the customer of a ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCustomer
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketCustomer'}->{'RequiredLock'} = ' '</code>

B.2.33. Frontend::Agent::Ticket::ViewEmailNew

B.2.33.1. Ticket::Frontend::AgentTicketEmail###Priority

Description	Value
Description:	Sets the default priority for new email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Priority'} = '3 normal'</code>

B.2.33.2. Ticket::Frontend::AgentTicketEmail###ArticleType

Description	Value
Description:	Sets the default article type for new email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'ArticleType'} = 'email'</code>

B.2.33.3. Ticket::Frontend::AgentTicketEmail###SenderType

Description	Value
Description:	Sets the default sender type for new email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'SenderType'} = 'agent'</code>

B.2.33.4. Ticket::Frontend::AgentTicketEmail###Subject

Description	Value
Description:	Sets the default subject for new email tickets (e.g. 'email Outbound') in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Subject'} = '';</code>

B.2.33.5. Ticket::Frontend::AgentTicketEmail###Body

Description	Value
Description:	Sets the default text for new email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Body'} = '';</code>

B.2.33.6. Ticket::Frontend::AgentTicketEmail###StateDefault

Description	Value
Description:	Sets the default next ticket state, after the creation of an email ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'StateDefault'} = 'open';</code>

B.2.33.7. Ticket::Frontend::AgentTicketEmail###StateType

Description	Value
Description:	Determines the next possible ticket states, after the creation of a new email ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'StateType'} = ['open', 'pending auto', 'pending reminder', 'closed'];</pre>

B.2.33.8. Ticket::Frontend::AgentTicketEmail###TicketFreeText

Description	Value
Description:	Ticket free text options shown in the email ticket screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'TicketFreeText'} = { '1' => '0', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0', '15' => '0', '16' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' };</pre>

B.2.33.9. Ticket::Frontend::AgentTicketEmail###TicketFreeTime

Description	Value
Description:	Ticket free time options shown in the email ticket screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'TicketFreeTime'} = { '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

B.2.33.10. Ticket::Frontend::AgentTicketEmail###ArticleFreeText

Description	Value
Description:	Article free text options shown in the email ticket screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'ArticleFreeText'} = { '1' => '0', '2' => '0', '3' => '0' };</pre>

B.2.33.11. Ticket::Frontend::AgentTicketEmail###HistoryType

Description	Value
Description:	Defines the history type for the email ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'HistoryType'} = 'Email'</pre>

B.2.33.12. Ticket::Frontend::AgentTicketEmail###HistoryComment

Description	Value
Description:	Defines the history comment for the email ticket screen action, which gets used for ticket history in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'HistoryComment'} = '';</code>

B.2.34. Frontend::Agent::Ticket::ViewEscalation

B.2.34.1.

Ticket::Frontend::AgentTicketEscalationView###ViewableTicketsPage

Description	Value
Description:	Shows all open tickets (even if they are locked) in the escalation view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'ViewableTicketsPage'}</code>

B.2.34.2. Ticket::Frontend::AgentTicketEscalationView###SortBy::Default

Description	Value
Description:	Defines the default ticket attribute for ticket sorting in the escalation view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'SortBy::Default'}</code>

B.2.34.3. Ticket::Frontend::AgentTicketEscalationView###Order::Default

Description	Value
Description:	Defines the default ticket order (after priority sort) in the escalation view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'Order::Default</code>

B.2.35. Frontend::Agent::Ticket::ViewForward**B.2.35.1. Ticket::Frontend::AgentTicketForward###Permission**

Description	Value
Description:	Required permissions to use the ticket forward screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketForward'}->{'Permission'} = 'for</code>

B.2.35.2. Ticket::Frontend::AgentTicketForward###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket forward screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketForward'}->{'RequiredLock'} = '1'</code>

B.2.35.3. Ticket::Frontend::AgentTicketForward###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after being forwarded, in the ticket forward screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketForward'}->{'StateDefault'} = 'c'</code>

B.2.35.4. Ticket::Frontend::AgentTicketForward###StateType

Description	Value
Description:	Defines the next possible states after forwarding a ticket in the ticket forward screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketForward'}->{'StateType'} = ['open', 'closed', 'pending reminder', 'pending auto'];</code>

B.2.35.5. Ticket::Frontend::AgentTicketForward###ArticleTypeDefault

Description	Value
Description:	Defines the default type of forwarded message in the ticket forward screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward

Description	Value
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketForward'}->{'ArticleTypeDefault'}</code>

B.2.35.6. Ticket::Frontend::AgentTicketForward###ArticleTypes

Description	Value
Description:	Specifies the different article types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketForward'}->{'ArticleTypes'} = ['email-external', 'email-internal'];</code>

B.2.35.7. Ticket::Frontend::AgentTicketForward###TicketFreeText

Description	Value
Description:	Ticket free text options shown in the ticket forward screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketForward'}->{'TicketFreeText'} = '1' => '0', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0', '15' => '0', '16' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' ;</code>

B.2.35.8. Ticket::Frontend::AgentTicketForward###TicketFreeTime

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

Description	Value
Description:	Ticket free time options shown in the ticket forward screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketForward'}->{'TicketFreeTime'} = '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

B.2.35.9. Ticket::Frontend::AgentTicketForward###ArticleFreeText

Description	Value
Description:	Article free text options shown in the ticket phone outbound screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketForward'}->{'ArticleFreeText'} = '1' => '0', '2' => '0', '3' => '0' };</pre>

B.2.36. Frontend::Agent::Ticket::ViewFreeText

B.2.36.1. Ticket::Frontend::AgentTicketFreeText###Permission

Description	Value
Description:	Required permissions to use the ticket free text screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Permission'} = 'rw'</code>

B.2.36.2. Ticket::Frontend::AgentTicketFreeText###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket free text screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'RequiredLock'} = '1'</code>

B.2.36.3. Ticket::Frontend::AgentTicketFreeText###TicketType

Description	Value
Description:	Sets the ticket type in the ticket free text screen of the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'TicketType'} = '1'</code>

B.2.36.4. Ticket::Frontend::AgentTicketFreeText###Service

Description	Value
Description:	Sets the service in the ticket free text screen of the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText

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

B.2.36.5. Ticket::Frontend::AgentTicketFreeText###Owner

Description	Value
Description:	Sets the ticket owner in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Owner'} = '0';</code>

B.2.36.6. Ticket::Frontend::AgentTicketFreeText###OwnerMandatory

Description	Value
Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'OwnerMandatory'} =</code>

B.2.36.7. Ticket::Frontend::AgentTicketFreeText###Responsible

Description	Value
Description:	Sets the responsible agent of the ticket in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1

Description	Value
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Responsible'} = '0'</code>

B.2.36.8. Ticket::Frontend::AgentTicketFreeText###State

Description	Value
Description:	If a note is added by an agent, sets the state of a ticket in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'State'} = '0';</code>

B.2.36.9. Ticket::Frontend::AgentTicketFreeText###StateType

Description	Value
Description:	Defines the next state of a ticket after adding a note, in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'StateType'} = ['open', 'closed', 'pending reminder', 'pending auto'];</code>

B.2.36.10. Ticket::Frontend::AgentTicketFreeText###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after adding a note, in the ticket free text screen of the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'StateDefault'} = ''</code>

B.2.36.11. Ticket::Frontend::AgentTicketFreeText###Note

Description	Value
Description:	Allows adding notes in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Note'} = ''</code>

B.2.36.12. Ticket::Frontend::AgentTicketFreeText###Subject

Description	Value
Description:	Defines the default subject of a note in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Subject'} = '\$Text'</code>

B.2.36.13. Ticket::Frontend::AgentTicketFreeText###Body

Description	Value
Description:	Defines the default body of a note in the ticket free text screen of the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Body'} = '';</code>

B.2.36.14. Ticket::Frontend::AgentTicketFreeText###InvolvedAgent

Description	Value
Description:	Shows a list of all the involved agents on this ticket, in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'InvolvedAgent'} =</code>

B.2.36.15. Ticket::Frontend::AgentTicketFreeText###InformAgent

Description	Value
Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'InformAgent'} = '0'</code>

B.2.36.16. Ticket::Frontend::AgentTicketFreeText###ArticleTypeDefault

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

Description	Value
Description:	Defines the default type of the note in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ArticleTypeDefault'</code>

B.2.36.17. Ticket::Frontend::AgentTicketFreeText###ArticleTypes

Description	Value
Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ArticleTypes'} = { 'note-external' => '1', 'note-internal' => '1', 'note-report' => '0' };</code>

B.2.36.18. Ticket::Frontend::AgentTicketFreeText###Priority

Description	Value
Description:	Shows the ticket priority options in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Priority'} = '0';</code>

B.2.36.19. Ticket::Frontend::AgentTicketFreeText###PriorityDefault

Description	Value
Description:	Defines the default ticket priority in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'PriorityDefault'} =</code>

B.2.36.20. Ticket::Frontend::AgentTicketFreeText###Title

Description	Value
Description:	Shows the title fields in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Title'} = '1';</code>

B.2.36.21. Ticket::Frontend::AgentTicketFreeText###TicketFreeText

Description	Value
Description:	Ticket free text options shown in the ticket free text screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'TicketFreeText'} = '1' => '1', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0', '15' => '0', '16' => '0', '2' => '1', '3' => '1', '4' => '1', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' };</code>

B.2.36.22. Ticket::Frontend::AgentTicketFreeText###TicketFreeTime

Description	Value
Description:	Ticket free time options shown in the ticket free text screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'TicketFreeTime'} = '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

B.2.36.23. Ticket::Frontend::AgentTicketFreeText###ArticleFreeText

Description	Value
Description:	Article free text options shown in the ticket free text screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ArticleFreeText'} = '1' => '0', '2' => '0', '3' => '0' };</pre>

B.2.36.24. Ticket::Frontend::AgentTicketFreeText###HistoryType

Description	Value
Description:	Defines the history type for the ticket free text screen action, which gets used for ticket history.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'HistoryType'} = 'A</pre>

B.2.36.25. Ticket::Frontend::AgentTicketFreeText###HistoryComment

Description	Value
Description:	Defines the history comment for the ticket free text screen action, which gets used for ticket history.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'HistoryComment'} =</code>

B.2.37. Frontend::Agent::Ticket::ViewHistory**B.2.37.1. Ticket::Frontend::HistoryOrder**

Description	Value
Description:	Shows the ticket history (reverse ordered) in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewHistory
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::HistoryOrder'} = 'normal';</code>

B.2.38. Frontend::Agent::Ticket::ViewMailbox**B.2.38.1. Ticket::Frontend::AgentTicketLockedView###SortBy::Default**

Description	Value
Description:	Defines the default ticket attribute for ticket sorting in the locked ticket view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketLockedView'}->{'SortBy::Default'}</code>

B.2.38.2. Ticket::Frontend::AgentTicketLockedView###Order::Default

Description	Value
Description:	Defines the default ticket order in the ticket locked view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketLockedView'}->{'Order::Default'}</code>

B.2.38.3.

Ticket::Frontend::AgentTicketResponsibleView###SortBy::Default

Description	Value
Description:	Defines the default ticket attribute for ticket sorting in the responsible view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsibleView'}->{'SortBy::Default'}</code>

B.2.38.4.

Ticket::Frontend::AgentTicketResponsibleView###Order::Default

Description	Value
Description:	Defines the default ticket order in the responsible view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsibleView'}->{'Order::Default'}</code>

B.2.38.5. Ticket::Frontend::AgentTicketWatchView###SortBy::Default

Description	Value
Description:	Defines the default ticket attribute for ticket sorting in the watch view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketWatchView'}->{'SortBy::Default'}</code>

B.2.38.6. Ticket::Frontend::AgentTicketWatchView###Order::Default

Description	Value
Description:	Defines the default ticket order in the watch view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketWatchView'}->{'Order::Default'} =</code>

B.2.39. Frontend::Agent::Ticket::ViewMerge

B.2.39.1. Ticket::Frontend::AgentTicketMerge###Permission

Description	Value
Description:	Required permissions to use the ticket merge screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'Permission'} = 'rw';</code>

B.2.39.2. Ticket::Frontend::AgentTicketMerge###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket merge screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'RequiredLock'} = '1';</code>

B.2.39.3. Ticket::Frontend::MergeText

Description	Value
Description:	When tickets are merged, the customer can be informed per email by setting the check box "Inform Sender". In this text area, you can define a pre-formatted text which can later be modified by the agents.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::MergeText'} = 'Your email with ticket number';</code>

B.2.39.4. Ticket::Frontend::AutomaticMergeText

Description	Value
Description:	When tickets are merged, a note will be added automatically to the ticket which is no longer active. In this text area you can define this text (This text cannot be changed by the agent).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AutomaticMergeText'} = 'Merged Ticket <OTRS</code>

B.2.40. Frontend::Agent::Ticket::ViewMove**B.2.40.1. Ticket::Frontend::MoveType**

Description	Value
Description:	Determines if the list of possible queues to move to ticket into should be displayed in a dropdown list or in a new window in the agent interface. If "New Window" is set you can add a move note to the ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::MoveType'} = 'form';</code>

B.2.40.2. Ticket::Frontend::AgentTicketMove###State

Description	Value
Description:	Allows to set a new ticket state in the move ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketMove'}->{'State'} = '1';</code>

B.2.40.3. Ticket::DefaultNextMoveStateType

Description	Value
Description:	Defines the next state of a ticket after being moved to another queue, in the move ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::DefaultNextMoveStateType'} = ['open', 'closed'];</code>

B.2.40.4. Ticket::Frontend::AgentTicketMove###Priority

Description	Value
Description:	Shows the ticket priority options in the move ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Priority'} = '0';</code>

B.2.40.5. Ticket::Frontend::AgentTicketMove###TicketFreeText

Description	Value
Description:	Ticket free text options shown in the move ticket screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketMove'}->{'TicketFreeText'} = { '1' => '0', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0', '15' => '0', '16' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' }; </pre>

B.2.40.6. Ticket::Frontend::AgentTicketMove###TicketFreeTime

Description	Value
Description:	Ticket free time options shown in the move ticket screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketMove'}->{'TicketFreeTime'} = { '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' }; </pre>

B.2.40.7. Ticket::Frontend::AgentTicketMove###Subject

Description	Value
Description:	Sets the default subject for notes added in the ticket move screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Subject'} = '\$Text{"Ch </pre>

B.2.40.8. Ticket::Frontend::AgentTicketMove###Body

Description	Value
Description:	Sets the default body text for notes added in the ticket move screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Body'} = '';</code>

B.2.41. Frontend::Agent::Ticket::ViewNote

B.2.41.1. Ticket::Frontend::AgentTicketNote###Permission

Description	Value
Description:	Required permissions to use the ticket note screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Permission'} = 'note';</code>

B.2.41.2. Ticket::Frontend::AgentTicketNote###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket note screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'RequiredLock'} = '0';</code>

B.2.41.3. Ticket::Frontend::AgentTicketNote###TicketType

Description	Value
Description:	Sets the ticket type in the ticket note screen of the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'TicketType'} = '0';</code>

B.2.41.4. Ticket::Frontend::AgentTicketNote###Service

Description	Value
Description:	Sets the service in the ticket note screen of the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Service'} = '0';</code>

B.2.41.5. Ticket::Frontend::AgentTicketNote###Owner

Description	Value
Description:	Sets the ticket owner in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Owner'} = '0';</code>

B.2.41.6. Ticket::Frontend::AgentTicketNote###OwnerMandatory

Description	Value
Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'OwnerMandatory'} = '0'</code>

B.2.41.7. Ticket::Frontend::AgentTicketNote###Responsible

Description	Value
Description:	Sets the responsible agent of the ticket in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Responsible'} = '0';</code>

B.2.41.8. Ticket::Frontend::AgentTicketNote###State

Description	Value
Description:	If a note is added by an agent, sets the state of a ticket in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'State'} = '0';</code>

B.2.41.9. Ticket::Frontend::AgentTicketNote###StateType

Description	Value
Description:	Defines the next state of a ticket after adding a note, in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'StateType'} = ['open', 'closed', 'pending reminder', 'pending auto'];</pre>

B.2.41.10. Ticket::Frontend::AgentTicketNote###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after adding a note, in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'StateDefault'} = 'open'</pre>

B.2.41.11. Ticket::Frontend::AgentTicketNote###Note

Description	Value
Description:	Allows adding notes in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Note'} = '1';</pre>

B.2.41.12. Ticket::Frontend::AgentTicketNote###Subject

Description	Value
Description:	Sets the default subject for notes added in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Subject'} = ' \$Text{"No</code>

B.2.41.13. Ticket::Frontend::AgentTicketNote###Body

Description	Value
Description:	Sets the default body text for notes added in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Body'} = '';</code>

B.2.41.14. Ticket::Frontend::AgentTicketNote###InvolvedAgent

Description	Value
Description:	Shows a list of all the involved agents on this ticket, in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'InvolvedAgent'} = '0';</code>

B.2.41.15. Ticket::Frontend::AgentTicketNote###InformAgent

Description	Value
Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'InformAgent'} = '0';</code>

B.2.41.16. Ticket::Frontend::AgentTicketNote###ArticleTypeDefault

Description	Value
Description:	Defines the default type of the note in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'ArticleTypeDefault'} =</code>

B.2.41.17. Ticket::Frontend::AgentTicketNote###ArticleTypes

Description	Value
Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'ArticleTypes'} = { 'note-external' => '1', 'note-internal' => '1', 'note-report' => '0' };</code>

B.2.41.18. Ticket::Frontend::AgentTicketNote###Priority

Description	Value
Description:	Shows the ticket priority options in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Priority'} = '0';</code>

B.2.41.19. Ticket::Frontend::AgentTicketNote###PriorityDefault

Description	Value
Description:	Defines the default ticket priority in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'PriorityDefault'} = '3';</code>

B.2.41.20. Ticket::Frontend::AgentTicketNote###Title

Description	Value
Description:	Shows the title fields in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Title'} = '0';</code>

B.2.41.21. Ticket::Frontend::AgentTicketNote###TicketFreeText

Description	Value
Description:	Ticket free text options shown in the ticket note screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'TicketFreeText'} = { '1' => '0', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0', '15' => '0', '16' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' };</pre>

B.2.41.22. Ticket::Frontend::AgentTicketNote###TicketFreeTime

Description	Value
Description:	Ticket free time options shown in the ticket note screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'TicketFreeTime'} = { '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

B.2.41.23. Ticket::Frontend::AgentTicketNote###ArticleFreeText

Description	Value
Description:	Article free text options shown in the ticket note screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'ArticleFreeText'} = { '1' => '0', '2' => '0', '3' => '0'};</pre>

B.2.41.24. Ticket::Frontend::AgentTicketNote###HistoryType

Description	Value
Description:	Defines the history type for the ticket note screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'HistoryType'} = 'AddNo</pre>

B.2.41.25. Ticket::Frontend::AgentTicketNote###HistoryComment

Description	Value
Description:	Defines the history comment for the ticket note screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'HistoryComment'} = '%%</pre>

B.2.42. Frontend::Agent::Ticket::ViewOwner

B.2.42.1. Ticket::Frontend::AgentTicketOwner###Permission

Description	Value
Description:	Required permissions to use the ticket owner screen of a zoomed ticket in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Permission'} = 'owner'</code>

B.2.42.2. Ticket::Frontend::AgentTicketOwner###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket owner screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'RequiredLock'} = '0';</code>

B.2.42.3. Ticket::Frontend::AgentTicketOwner###TicketType

Description	Value
Description:	Sets the ticket type in the ticket owner screen of a zoomed ticket in the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'TicketType'} = '0';</code>

B.2.42.4. Ticket::Frontend::AgentTicketOwner###Service

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

Description	Value
Description:	Sets the service in the ticket owner screen of a zoomed ticket in the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Service'} = '0';</code>

B.2.42.5. Ticket::Frontend::AgentTicketOwner###Owner

Description	Value
Description:	Sets the ticket owner in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Owner'} = '1';</code>

B.2.42.6. Ticket::Frontend::AgentTicketOwner###OwnerMandatory

Description	Value
Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'OwnerMandatory'} = '1';</code>

B.2.42.7. Ticket::Frontend::AgentTicketOwner###Responsible

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

Description	Value
Description:	Sets the responsible agent of the ticket in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Responsible'} = '0';</code>

B.2.42.8. Ticket::Frontend::AgentTicketOwner###State

Description	Value
Description:	If a note is added by an agent, sets the state of the ticket in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'State'} = '0';</code>

B.2.42.9. Ticket::Frontend::AgentTicketOwner###StateType

Description	Value
Description:	Defines the next state of a ticket after adding a note, in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'StateType'} = ['open', 'pending reminder', 'pending auto'];</code>

B.2.42.10. Ticket::Frontend::AgentTicketOwner###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after adding a note, in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'StateDefault'} = 'open'</code>

B.2.42.11. Ticket::Frontend::AgentTicketOwner###Note

Description	Value
Description:	Allows adding notes in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Note'} = '1';</code>

B.2.42.12. Ticket::Frontend::AgentTicketOwner###Subject

Description	Value
Description:	Sets the default subject for notes added in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Subject'} = '\$Text{"O</code>

B.2.42.13. Ticket::Frontend::AgentTicketOwner###Body

Description	Value
Description:	Sets the default body text for notes added in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Body'} = '';</code>

B.2.42.14. Ticket::Frontend::AgentTicketOwner###InvolvedAgent

Description	Value
Description:	Shows a list of all the involved agents on this ticket, in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'InvolvedAgent'} = '0';</code>

B.2.42.15. Ticket::Frontend::AgentTicketOwner###InformAgent

Description	Value
Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'InformAgent'} = '0';</code>

B.2.42.16. Ticket::Frontend::AgentTicketOwner###ArticleTypeDefault

Description	Value
Description:	Defines the default type of the note in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'ArticleTypeDefault'} =</code>

B.2.42.17. Ticket::Frontend::AgentTicketOwner###ArticleTypes

Description	Value
Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };</code>

B.2.42.18. Ticket::Frontend::AgentTicketOwner###Priority

Description	Value
Description:	Shows the ticket priority options in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Priority'} = '0';</code>

B.2.42.19. Ticket::Frontend::AgentTicketOwner###PriorityDefault

Description	Value
Description:	Defines the default ticket priority in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'PriorityDefault'} =</code>

B.2.42.20. Ticket::Frontend::AgentTicketOwner###Title

Description	Value
Description:	Shows the title fields in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Title'} =</code>

B.2.42.21. Ticket::Frontend::AgentTicketOwner###TicketFreeText

Description	Value
Description:	Ticket free text options shown in the ticket owner screen of a zoomed ticket in the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'TicketFreeText'} = { '1' => '0', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0', '15' => '0', '16' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' };</pre>

B.2.42.22. Ticket::Frontend::AgentTicketOwner###TicketFreeTime

Description	Value
Description:	Ticket free time options shown in the ticket owner screen of a zoomed ticket in the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'TicketFreeTime'} = { '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

B.2.42.23. Ticket::Frontend::AgentTicketOwner###ArticleFreeText

Description	Value
Description:	Article free text options shown in the ticket owner screen of a zoomed ticket in the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'ArticleFreeText'} = { '1' => '0', '2' => '0', '3' => '0' };</pre>

B.2.42.24. Ticket::Frontend::AgentTicketOwner###HistoryType

Description	Value
Description:	Defines the history type for the ticket owner screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'HistoryType'} = 'AddN</code>

B.2.42.25. Ticket::Frontend::AgentTicketOwner###HistoryComment

Description	Value
Description:	Defines the history comment for the ticket owner screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'HistoryComment'} = '%</code>

B.2.43. Frontend::Agent::Ticket::ViewPending

B.2.43.1. Ticket::Frontend::AgentTicketPending###Permission

Description	Value
Description:	Required permissions to use the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Permission'} = 'pen</code>

B.2.43.2. Ticket::Frontend::AgentTicketPending###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket pending screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'RequiredLock'} = '1'</code>

B.2.43.3. Ticket::Frontend::AgentTicketPending###TicketType

Description	Value
Description:	Sets the ticket type in the ticket pending screen of a zoomed ticket in the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'TicketType'} = '0';</code>

B.2.43.4. Ticket::Frontend::AgentTicketPending###Service

Description	Value
Description:	Sets the service in the ticket pending screen of a zoomed ticket in the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Service'} = '0';</code>

B.2.43.5. Ticket::Frontend::AgentTicketPending###Owner

Description	Value
Description:	Sets the ticket owner in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Owner'} = '0';</code>

B.2.43.6. Ticket::Frontend::AgentTicketPending###OwnerMandatory

Description	Value
Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'OwnerMandatory'} =</code>

B.2.43.7. Ticket::Frontend::AgentTicketPending###Responsible

Description	Value
Description:	Sets the responsible agent of the ticket in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Responsible'} = '0'</code>

B.2.43.8. Ticket::Frontend::AgentTicketPending###State

Description	Value
Description:	If a note is added by an agent, sets the state of the ticket in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'State'} = '1';</code>

B.2.43.9. Ticket::Frontend::AgentTicketPending###StateType

Description	Value
Description:	Defines the next state of a ticket after adding a note, in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'StateType'} = ['pending reminder', 'pending auto'];</code>

B.2.43.10. Ticket::Frontend::AgentTicketPending###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after adding a note, in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'StateDefault'} = 'p</code>

B.2.43.11. Ticket::Frontend::AgentTicketPending###Note

Description	Value
Description:	Allows adding notes in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Note'} = '1';</code>

B.2.43.12. Ticket::Frontend::AgentTicketPending###Subject

Description	Value
Description:	Sets the default subject for notes added in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Subject'} = '\$Text{'</code>

B.2.43.13. Ticket::Frontend::AgentTicketPending###Body

Description	Value
Description:	Sets the default body text for notes added in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Body'} = '';</code>

B.2.43.14. Ticket::Frontend::AgentTicketPending###InvolvedAgent

Description	Value
Description:	Shows a list of all the involved agents on this ticket, in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'InvolvedAgent'} = '0'</code>

B.2.43.15. Ticket::Frontend::AgentTicketPending###InformAgent

Description	Value
Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'InformAgent'} = '0'</code>

B.2.43.16. Ticket::Frontend::AgentTicketPending###ArticleTypeDefault

Description	Value
Description:	Defines the default type of the note in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'ArticleTypeDefault'}</code>

B.2.43.17. Ticket::Frontend::AgentTicketPending###ArticleTypes

Description	Value
Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };</pre>

B.2.43.18. Ticket::Frontend::AgentTicketPending###Priority

Description	Value
Description:	Shows the ticket priority options in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Priority'} = '0';</pre>

B.2.43.19. Ticket::Frontend::AgentTicketPending###PriorityDefault

Description	Value
Description:	Defines the default ticket priority in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'PriorityDefault'} =</pre>

B.2.43.20. Ticket::Frontend::AgentTicketPending###Title

Description	Value
Description:	Shows the title fields in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Title'} = '0';</code>

B.2.43.21. Ticket::Frontend::AgentTicketPending###TicketFreeText

Description	Value
Description:	Ticket free text options shown in the ticket pending screen of a zoomed ticket in the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'TicketFreeText'} = '1' => '0', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0', '15' => '0', '16' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' };</code>

B.2.43.22. Ticket::Frontend::AgentTicketPending###TicketFreeTime

Description	Value
Description:	Ticket free time options shown in the ticket pending screen of a zoomed ticket in the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'TicketFreeTime'} = '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

B.2.43.23. Ticket::Frontend::AgentTicketPending###ArticleFreeText

Description	Value
Description:	Article free text options shown in the ticket pending screen of a zoomed ticket in the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'ArticleFreeText'} = '1' => '0', '2' => '0', '3' => '0' };</pre>

B.2.43.24. Ticket::Frontend::AgentTicketPending###HistoryType

Description	Value
Description:	Defines the history type for the ticket pending screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'HistoryType'} = 'Ad</pre>

B.2.43.25. Ticket::Frontend::AgentTicketPending###HistoryComment

Description	Value
Description:	Defines the history comment for the ticket pending screen action, which gets used for ticket history in the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'HistoryComment'} =</code>

B.2.44. Frontend::Agent::Ticket::ViewPhoneNew

B.2.44.1. Ticket::Frontend::AgentTicketPhone###Priority

Description	Value
Description:	Sets the default priority for new phone tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'Priority'} = '3 normal'</code>

B.2.44.2. Ticket::Frontend::AgentTicketPhone###ArticleType

Description	Value
Description:	Sets the default article type for new phone tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'ArticleType'} = 'phone'</code>

B.2.44.3. Ticket::Frontend::AgentTicketPhone###SenderType

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

Description	Value
Description:	Sets the default sender type for new phone ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'SenderType'} = 'custo</code>

B.2.44.4. Ticket::Frontend::AgentTicketPhone###Subject

Description	Value
Description:	Sets the default subject for new phone tickets (e.g. 'Phone call') in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'Subject'} = '';</code>

B.2.44.5. Ticket::Frontend::AgentTicketPhone###Body

Description	Value
Description:	Sets the default note text for new telephone tickets. E.g 'New ticket via call' in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'Body'} = '';</code>

B.2.44.6. Ticket::Frontend::AgentTicketPhone###StateDefault

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

Description	Value
Description:	Sets the default next state for new phone tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'StateDefault'} = 'open'</code>

B.2.44.7. Ticket::Frontend::AgentTicketPhone###StateType

Description	Value
Description:	Determines the next possible ticket states, after the creation of a new phone ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'StateType'} = ['open', 'pending auto', 'pending reminder', 'closed'];</code>

B.2.44.8. Ticket::Frontend::AgentTicketPhone###TicketFreeText

Description	Value
Description:	Ticket free text options shown in the phone ticket screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'TicketFreeText'} = { '1' => '0', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0', '15' => '0', '16' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' };</code>

B.2.44.9. Ticket::Frontend::AgentTicketPhone###TicketFreeTime

Description	Value
Description:	Ticket free time options shown in the phone ticket screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'TicketFreeTime'} = { '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' }; </pre>

B.2.44.10. Ticket::Frontend::AgentTicketPhone###ArticleFreeText

Description	Value
Description:	Article free text options shown in the phone ticket screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'ArticleFreeText'} = { '1' => '0', '2' => '0', '3' => '0' }; </pre>

B.2.44.11. Ticket::Frontend::AgentTicketPhone###HistoryType

Description	Value
Description:	Defines the history type for the phone ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'HistoryType'} = 'Phone </pre>

B.2.44.12. Ticket::Frontend::AgentTicketPhone###HistoryComment

Description	Value
Description:	Defines the history comment for the phone ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'HistoryComment'} = "";</code>

B.2.44.13. Ticket::Frontend::AgentTicketPhone###SplitLinkType

Description	Value
Description:	Sets the default link type of splitted tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'SplitLinkType'} = { 'Direction' => 'Target', 'LinkType' => 'ParentChild'};</code>

B.2.45. Frontend::Agent::Ticket::ViewPhoneOutbound**B.2.45.1. Ticket::Frontend::AgentTicketPhoneOutbound###Permission**

Description	Value
Description:	Required permissions to use the ticket phone outbound screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'Permission'} =</code>

B.2.45.2. Ticket::Frontend::AgentTicketPhoneOutbound###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket phone outbound screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'RequiredLock'}</code>

B.2.45.3. Ticket::Frontend::AgentTicketPhoneOutbound###ArticleType

Description	Value
Description:	Defines the default type of the note in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'ArticleType'}</code>

B.2.45.4. Ticket::Frontend::AgentTicketPhoneOutbound###SenderType

Description	Value
Description:	Defines the default sender type for phone tickets in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'SenderType'} =</code>

B.2.45.5. Ticket::Frontend::AgentTicketPhoneOutbound###Subject

Description	Value
Description:	Defines the default subject for phone tickets in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'Subject'} =</code>

B.2.45.6. Ticket::Frontend::AgentTicketPhoneOutbound###Body

Description	Value
Description:	Defines the default note body text for phone tickets in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'Body'} =</code>

B.2.45.7. Ticket::Frontend::AgentTicketPhoneOutbound###State

Description	Value
Description:	Defines the default ticket next state after adding a phone note in the ticket phone outbound screen of the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'State'} = 'cl</code>

B.2.45.8. Ticket::Frontend::AgentTicketPhoneOutbound###StateType

Description	Value
Description:	Next possible ticket states after adding a phone note in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'StateType'} = 'open', 'pending auto', 'pending reminder', 'closed'];</code>

B.2.45.9.

Ticket::Frontend::AgentTicketPhoneOutbound###TicketFreeText

Description	Value
Description:	Ticket free text options shown in the ticket phone outbound screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'TicketFreeText' '1' => '0', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0', '15' => '0', '16' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' };</code>

B.2.45.10.**Ticket::Frontend::AgentTicketPhoneOutbound###TicketFreeTime**

Description	Value
Description:	Ticket free time options shown in the ticket phone outbound screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'TicketFreeTime' '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

B.2.45.11.**Ticket::Frontend::AgentTicketPhoneOutbound###ArticleFreeText**

Description	Value
Description:	Article free text options shown in the ticket phone outbound screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'ArticleFreeText' '1' => '0', '2' => '0', '3' => '0' };</pre>

B.2.45.12. Ticket::Frontend::AgentTicketPhoneOutbound###HistoryType

Description	Value
Description:	Defines the history type for the ticket phone outbound screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'HistoryType'}</code>

B.2.45.13.**Ticket::Frontend::AgentTicketPhoneOutbound###HistoryComment**

Description	Value
Description:	Defines the history comment for the ticket phone outbound screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'HistoryComment'}</code>

B.2.46. Frontend::Agent::Ticket::ViewPriority**B.2.46.1. Ticket::Frontend::AgentTicketPriority###Permission**

Description	Value
Description:	Required permissions to use the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Permission'} = 'pr</code>

B.2.46.2. Ticket::Frontend::AgentTicketPriority###RequiredLock

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

Description	Value
Description:	Defines if a ticket lock is required in the ticket priority screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'RequiredLock'} = '0'</code>

B.2.46.3. Ticket::Frontend::AgentTicketPriority###TicketType

Description	Value
Description:	Sets the ticket type in the ticket priority screen of a zoomed ticket in the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'TicketType'} = '0'</code>

B.2.46.4. Ticket::Frontend::AgentTicketPriority###Service

Description	Value
Description:	Sets the service in the ticket priority screen of a zoomed ticket in the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Service'} = '0';</code>

B.2.46.5. Ticket::Frontend::AgentTicketPriority###Owner

Description	Value
Description:	Sets the ticket owner in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Owner'} = '0';</code>

B.2.46.6. Ticket::Frontend::AgentTicketPriority###OwnerMandatory

Description	Value
Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'OwnerMandatory'} =</code>

B.2.46.7. Ticket::Frontend::AgentTicketPriority###Responsible

Description	Value
Description:	Sets the responsible agent of the ticket in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Responsible'} = '0'</code>

B.2.46.8. Ticket::Frontend::AgentTicketPriority###State

Description	Value
Description:	If a note is added by an agent, sets the state of the ticket in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'State'} = '0';</code>

B.2.46.9. Ticket::Frontend::AgentTicketPriority###StateType

Description	Value
Description:	Defines the next state of a ticket after adding a note, in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'StateType'} = ['open', 'pending reminder', 'pending auto'];</code>

B.2.46.10. Ticket::Frontend::AgentTicketPriority###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after adding a note, in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'StateDefault'} = 'open';</code>

B.2.46.11. Ticket::Frontend::AgentTicketPriority###Note

Description	Value
Description:	Allows adding notes in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Note'} = '1';</code>

B.2.46.12. Ticket::Frontend::AgentTicketPriority###Subject

Description	Value
Description:	Sets the default subject for notes added in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Subject'} = '\$Text';</code>

B.2.46.13. Ticket::Frontend::AgentTicketPriority###Body

Description	Value
Description:	Sets the default body text for notes added in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Body'} = '';</code>

B.2.46.14. Ticket::Frontend::AgentTicketPriority###InvolvedAgent

Description	Value
Description:	Shows a list of all the involved agents on this ticket, in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'InvolvedAgent'} =</code>

B.2.46.15. Ticket::Frontend::AgentTicketPriority###InformAgent

Description	Value
Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'InformAgent'} = '0</code>

B.2.46.16. Ticket::Frontend::AgentTicketPriority###ArticleTypeDefault

Description	Value
Description:	Defines the default type of the note in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ArticleTypeDefault'</code>

B.2.46.17. Ticket::Frontend::AgentTicketPriority###ArticleTypes

Description	Value
Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };</pre>

B.2.46.18. Ticket::Frontend::AgentTicketPriority###Priority

Description	Value
Description:	Shows the ticket priority options in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Priority'} = '1';</pre>

B.2.46.19. Ticket::Frontend::AgentTicketPriority###PriorityDefault

Description	Value
Description:	Defines the default ticket priority in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'PriorityDefault'} =</pre>

B.2.46.20. Ticket::Frontend::AgentTicketPriority###Title

Description	Value
Description:	Shows the title fields in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Title'} = '0';</code>

B.2.46.21. Ticket::Frontend::AgentTicketPriority###TicketFreeText

Description	Value
Description:	Ticket free text options shown in the ticket priority screen of a zoomed ticket in the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'TicketFreeText'} = '1' => '0', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0', '15' => '0', '16' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' };</code>

B.2.46.22. Ticket::Frontend::AgentTicketPriority###TicketFreeTime

Description	Value
Description:	Ticket free time options shown in the ticket priority screen of a zoomed ticket in the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'TicketFreeTime'} = '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

B.2.46.23. Ticket::Frontend::AgentTicketPriority###ArticleFreeText

Description	Value
Description:	Article free text options shown in the ticket priority screen of a zoomed ticket in the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ArticleFreeText'} = '1' => '0', '2' => '0', '3' => '0' };</pre>

B.2.46.24. Ticket::Frontend::AgentTicketPriority###HistoryType

Description	Value
Description:	Defines the history type for the ticket priority screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'HistoryType'} = 'A</pre>

B.2.46.25. Ticket::Frontend::AgentTicketPriority###HistoryComment

Description	Value
Description:	Defines the history comment for the ticket priority screen action, which gets used for ticket history in the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'HistoryComment'} =</code>

B.2.47. Frontend::Agent::Ticket::ViewQueue

B.2.47.1. Ticket::Frontend::AgentTicketQueue###StripEmptyLines

Description	Value
Description:	Strips empty lines on the ticket preview in the queue view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'StripEmptyLines'} =</code>

B.2.47.2. Ticket::Frontend::AgentTicketQueue###ViewAllPossibleTickets

Description	Value
Description:	Shows all both ro and rw queues in the queue view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'ViewAllPossibleTickets</code>

B.2.47.3. Ticket::Frontend::AgentTicketQueue###HighlightAge1

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

Description	Value
Description:	Sets the age in minutes (first level) for highlighting queues that contain untouched tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'HighlightAge1'} = '14'</code>

B.2.47.4. Ticket::Frontend::AgentTicketQueue###HighlightAge2

Description	Value
Description:	Sets the age in minutes (second level) for highlighting queues that contain untouched tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'HighlightAge2'} = '28'</code>

B.2.47.5. Ticket::Frontend::AgentTicketQueue###Blink

Description	Value
Description:	Activates a blinking mechanism of the queue that contains the oldest ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'Blink'} = '1';</code>

B.2.47.6. Ticket::Frontend::AgentTicketQueue###QueueSort

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

Description	Value
Description:	Sorts the tickets (ascendingly or descendingly) when a single queue is selected in the queue view and after the tickets are sorted by priority. Values: 0 = ascending (oldest on top, default), 1 = descending (youngest on top). Use the QueueID for the key and 0 or 1 for value.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'QueueSort'} = { '3' => '0', '7' => '1' };</pre>

B.2.47.7. Ticket::Frontend::AgentTicketQueue###SortBy::Default

Description	Value
Description:	Defines the default sort criteria for all queues displayed in the queue view, after sort by priority is done.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'SortBy::Default'} = '</pre>

B.2.47.8. Ticket::Frontend::AgentTicketQueue###Order::Default

Description	Value
Description:	Defines the default sort order for all queues in the queue view, after priority sort.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'Order::Default'} = 'U</pre>

B.2.48. Frontend::Agent::Ticket::ViewResponsible

B.2.48.1. Ticket::Frontend::AgentTicketResponsible###Permission

Description	Value
Description:	Required permissions to use the ticket responsible screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Permission'} =</code>

B.2.48.2. Ticket::Frontend::AgentTicketResponsible###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket responsible screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'RequiredLock'} =</code>

B.2.48.3. Ticket::Frontend::AgentTicketResponsible###TicketType

Description	Value
Description:	Sets the ticket type in the ticket responsible screen of the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'TicketType'} =</code>

B.2.48.4. Ticket::Frontend::AgentTicketResponsible###Service

Description	Value
Description:	Sets the service in the ticket responsible screen of the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Service'} = '0'</code>

B.2.48.5. Ticket::Frontend::AgentTicketResponsible###Owner

Description	Value
Description:	Sets the ticket owner in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Owner'} = '0';</code>

B.2.48.6. Ticket::Frontend::AgentTicketResponsible###OwnerMandatory

Description	Value
Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'OwnerMandatory'}</code>

B.2.48.7. Ticket::Frontend::AgentTicketResponsible###Responsible

Description	Value
Description:	Sets the responsible agent of the ticket in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Responsible'} =</code>

B.2.48.8. Ticket::Frontend::AgentTicketResponsible###State

Description	Value
Description:	If a note is added by an agent, sets the state of a ticket in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'State'} = '0';</code>

B.2.48.9. Ticket::Frontend::AgentTicketResponsible###StateType

Description	Value
Description:	Defines the next state of a ticket after adding a note, in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'StateType'} = ['open', 'pending reminder', 'pending auto'];</code>

B.2.48.10. Ticket::Frontend::AgentTicketResponsible###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after adding a note, in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'StateDefault'} =</code>

B.2.48.11. Ticket::Frontend::AgentTicketResponsible###Note

Description	Value
Description:	Allows adding notes in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Note'} = '1';</code>

B.2.48.12. Ticket::Frontend::AgentTicketResponsible###Subject

Description	Value
Description:	Sets the default subject for notes added in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1

Description	Value
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Subject'} = '\$T</code>

B.2.48.13. Ticket::Frontend::AgentTicketResponsible###Body

Description	Value
Description:	Sets the default body text for notes added in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Body'} = '';</code>

B.2.48.14. Ticket::Frontend::AgentTicketResponsible###InvolvedAgent

Description	Value
Description:	Shows a list of all the involved agents on this ticket, in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'InvolvedAgent'}</code>

B.2.48.15. Ticket::Frontend::AgentTicketResponsible###InformAgent

Description	Value
Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket responsible screen of the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'InformAgent'} =</code>

B.2.48.16.**Ticket::Frontend::AgentTicketResponsible###ArticleTypeDefault**

Description	Value
Description:	Defines the default type of the note in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'ArticleTypeDefault'}</code>

B.2.48.17. Ticket::Frontend::AgentTicketResponsible###ArticleTypes

Description	Value
Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'ArticleTypes'} = 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };</code>

B.2.48.18. Ticket::Frontend::AgentTicketResponsible###Priority

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

Description	Value
Description:	Shows the ticket priority options in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Priority'} = '0'</code>

B.2.48.19. Ticket::Frontend::AgentTicketResponsible###PriorityDefault

Description	Value
Description:	Defines the default ticket priority in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'PriorityDefault'}</code>

B.2.48.20. Ticket::Frontend::AgentTicketResponsible###Title

Description	Value
Description:	Shows the title fields in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Title'} = '1';</code>

B.2.48.21. Ticket::Frontend::AgentTicketResponsible###TicketFreeText

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

Description	Value
Description:	Ticket free text options shown in the ticket responsible screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'TicketFreeText'} '1' => '0', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0', '15' => '0', '16' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' };</pre>

B.2.48.22. Ticket::Frontend::AgentTicketResponsible###TicketFreeTime

Description	Value
Description:	Ticket free time options shown in the ticket responsible screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'TicketFreeTime'} '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

B.2.48.23. Ticket::Frontend::AgentTicketResponsible###ArticleFreeText

Description	Value
Description:	Article free text options shown in the ticket responsible screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'ArticleFreeText'}</code> <code>'1' => '0', '2' => '0', '3' => '0' };</code>

B.2.48.24. Ticket::Frontend::AgentTicketResponsible###HistoryType

Description	Value
Description:	Defines the history type for the ticket responsible screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'HistoryType'}</code> =

B.2.48.25. Ticket::Frontend::AgentTicketResponsible###HistoryComment

Description	Value
Description:	Defines the history comment for the ticket responsible screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'HistoryComment'}</code>

B.2.49. Frontend::Agent::Ticket::ViewSearch

B.2.49.1.

Ticket::Frontend::AgentTicketSearch###ExtendedSearchCondition

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

Description	Value
Description:	Allows extended search conditions in ticket search of the agent interface. With this feature you can search w. g. with this kind of conditions like "(key1&&key2)" or "(key1 key2)".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'ExtendedSearchCondi...</code>

B.2.49.2. Ticket::Frontend::AgentTicketSearch###SearchLimit

Description	Value
Description:	Maximum number of tickets to be displayed in the result of a search in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchLimit'} = '200</code>

B.2.49.3. Ticket::Frontend::AgentTicketSearch###SearchPageShown

Description	Value
Description:	Number of tickets to be displayed in each page of a search result in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchPageShown'} =</code>

B.2.49.4.

Ticket::Frontend::AgentTicketSearch###SearchViewableTicketLines

Description	Value
Description:	Number of lines (per ticket) that are shown by the search utility in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchViewableTicketLines'}</code>

B.2.49.5. Ticket::Frontend::AgentTicketSearch###TicketFreeText

Description	Value
Description:	Ticket free text options shown in the ticket search of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'TicketFreeText'} = { '1' => '1', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0', '15' => '0', '16' => '0', '2' => '1', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0'};</code>

B.2.49.6. Ticket::Frontend::AgentTicketSearch###TicketFreeTime

Description	Value
Description:	Ticket free time options shown in the ticket search of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'TicketFreeTime'} = { '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

B.2.49.7. Ticket::Frontend::AgentTicketSearch###SortBy::Default

Description	Value
Description:	Defines the default ticket attribute for ticket sorting of the ticket search result of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SortBy::Default'} =</pre>

B.2.49.8. Ticket::Frontend::AgentTicketSearch###Order::Default

Description	Value
Description:	Defines the default ticket order in the ticket search result of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Order::Default'} =</pre>

B.2.49.9. Ticket::Frontend::AgentTicketSearch###SearchArticleCSVTree

Description	Value
Description:	Exports the whole article tree in search result (it can affect the system performance).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchArticleCSVTree'</code>

B.2.49.10. Ticket::Frontend::AgentTicketSearch###SearchCSVData

Description	Value
Description:	Data used to export the search result in CSV format.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchCSVData'} = ['TicketNumber', 'Age', 'Created', 'Closed', 'FirstLock', 'FirstResponse', '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', 'TicketFreeTime3', 'TicketFreeTime4', 'TicketFreeTime5', 'TicketFreeTime6', 'ArticleTree', 'SolutionInMin', 'SolutionDiffInMin', 'FirstResponseInMin', 'FirstResponseDiffInMin'];</code>

B.2.49.11. Ticket::Frontend::AgentTicketSearch###ArticleCreateTime

Description	Value
Description:	Includes article create times in the ticket search of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch

Description	Value
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'ArticleCreateTime'}=</code>

B.2.49.12. Ticket::Frontend::AgentTicketSearch###Defaults###Fulltext

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Fulltext'}</code>

B.2.49.13.

Ticket::Frontend::AgentTicketSearch###Defaults###TicketNumber

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketNumber'}</code>

B.2.49.14. Ticket::Frontend::AgentTicketSearch###Defaults###Title

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0

Description	Value
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Title'}</code>

B.2.49.15. Ticket::Frontend::AgentTicketSearch###Defaults###From

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'From'}</code>

B.2.49.16. Ticket::Frontend::AgentTicketSearch###Defaults###To

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'To'} =</code>

B.2.49.17. Ticket::Frontend::AgentTicketSearch###Defaults###Cc

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Cc'} =</code>

B.2.49.18. Ticket::Frontend::AgentTicketSearch###Defaults###Subject

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Subject'}</code>

B.2.49.19. Ticket::Frontend::AgentTicketSearch###Defaults###Body

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Body'}</code>

B.2.49.20.

Ticket::Frontend::AgentTicketSearch###Defaults###CustomerID

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'CustomerUserLogin'}</code>

B.2.49.21.**Ticket::Frontend::AgentTicketSearch###Defaults###CustomerUserLogin**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'CustomerUserLogin'}</code>

B.2.49.22. Ticket::Frontend::AgentTicketSearch###Defaults###StateIDs

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'StateIDs'}</code>

B.2.49.23. Ticket::Frontend::AgentTicketSearch###Defaults###QueueIDs

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'QueueID'}</code>

B.2.49.24.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketCreateTimePoint**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCreateTimePoint'}</code>

B.2.49.25.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketCreateTimeSlot**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCreateTimeSlot'}</code>

B.2.49.26.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketChangeTimePoint**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0

Description	Value
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketC</code>

B.2.49.27.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketChangeTimeSlot**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketC</code>

B.2.49.28.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketCloseTimePoint**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketC</code>

B.2.49.29.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketCloseTimeSlot**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch

Description	Value
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketC</code>

B.2.49.30.**Ticket::Frontend::AgentTicketSearch###Defaults###ArticleCreateTimePoint**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Article</code>

B.2.49.31.**Ticket::Frontend::AgentTicketSearch###Defaults###ArticleCreateTimeSlot**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Article</code>

B.2.49.32.**Ticket::Frontend::AgentTicketSearch###Defaults###SearchInArchive**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'SearchI</code>

B.2.49.33.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeTime1**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.34.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeTime2**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.35.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeTime3**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.36.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeTime4**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.37.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeTime5**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.38.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeText1**

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

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.39.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeText2**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.40.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeText3**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.41.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeText4**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.42.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeText5**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.43.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeText6**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.44.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeText7**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.45.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeText8**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.46.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeText9**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.47.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeText10**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.48.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeText11**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.49.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeText12**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.50.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeText13**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.51.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeText14**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.52.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeText15**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.53.**Ticket::Frontend::AgentTicketSearch###Defaults###TicketFreeText16**

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketF</code>

B.2.49.54.**Ticket::Frontend::CustomerTicketSearch###SearchArticleCSVTree**

Description	Value
Description:	Exports the whole article tree in search result (it can affect the system performance).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchArticleCSVTr</code>

B.2.50. Frontend::Agent::Ticket::ViewStatus**B.2.50.1.****Ticket::Frontend::AgentTicketStatusView###ViewableTicketsPage**

Description	Value
Description:	Shows all open tickets (even if they are locked) in the status view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'ViewableTicketsPa</code>

B.2.50.2. Ticket::Frontend::AgentTicketStatusView###SortBy::Default

Description	Value
Description:	Defines the default ticket attribute for ticket sorting in the status view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'SortBy::Default'}</code>

B.2.50.3. Ticket::Frontend::AgentTicketStatusView###Order::Default

Description	Value
Description:	Defines the default ticket order (after priority sort) in the status view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'Order::Default'}</code>

B.2.51. Frontend::Agent::Ticket::ViewZoom**B.2.51.1. Ticket::Frontend::PlainView**

Description	Value
Description:	Shows a link to see a zoomed email ticket in plain text.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PlainView'} = '0';</code>

B.2.51.2. Ticket::Frontend::ZoomExpand

Description	Value
Description:	Shows all the articles of the ticket (expanded) in the zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::ZoomExpand'} = '0';</code>

B.2.51.3. Ticket::Frontend::ZoomExpandSort

Description	Value
Description:	Shows the articles sorted normally or in reverse, under ticket zoom in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::ZoomExpandSort'} = 'normal';</code>

B.2.51.4. Ticket::ZoomAttachmentDisplayCount

Description	Value
Description:	Shows a count of icons in the ticket zoom, if the article has attachments.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::ZoomAttachmentDisplayCount'} = '20';</code>

B.2.51.5. Ticket::ZoomTimeDisplay

Description	Value
Description:	Displays the accounted time for an article in the ticket zoom view.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::ZoomTimeDisplay'} = '0';</code>

B.2.51.6. Ticket::Frontend::TicketArticleFilter

Description	Value
Description:	Activates the article filter in the zoom view to specify which articles should be shown.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::TicketArticleFilter'} = '0';</code>

B.2.51.7. Ticket::Frontend::HTMLArticleHeightDefault

Description	Value
Description:	Set the default height (in pixels) of inline HTML articles in AgentTicketZoom.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::HTMLArticleHeightDefault'} = '100';</code>

B.2.51.8. Ticket::Frontend::HTMLArticleHeightMax

Description	Value
Description:	Set the maximum height (in pixels) of inline HTML articles in AgentTicketZoom.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::HTMLArticleHeightMax'} = '2500';</code>

B.2.52. Frontend::Agent::TicketOverview

B.2.52.1. Ticket::Frontend::Overview###Small

Description	Value
Description:	Allows having a small format ticket overview (CustomerInfo => 1 - shows also the customer information).
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::Overview'}->{'Small'} = { 'CustomerInfo' => '1', 'Module' => 'Kernel::Output::HTML::TicketOverviewSmall', 'Name' => 'Small', 'NameShort' => 'S' };</code>

B.2.52.2. Ticket::Frontend::OverviewSmall###ColumnHeader

Description	Value
Description:	Shows either the last customer article's subject or the ticket title in the small format overview.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::OverviewSmall'}->{'ColumnHeader'} = 'LastCu</code>

B.2.52.3. Ticket::Frontend::Overview###Medium

Description	Value
Description:	Allows having a medium format ticket overview (CustomerInfo => 1 - shows also the customer information).
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::Overview'}->{'Medium'} = { 'CustomerInfo' => '0', 'Module' => 'Kernel::Output::HTML::TicketOverviewMedium', 'Name' => 'Medium', 'NameShort' => 'M', 'TicketActionsPerTicket' => '0' };</pre>

B.2.52.4. Ticket::Frontend::Overview###Preview

Description	Value
Description:	Shows a preview of the ticket overview (CustomerInfo => 1 - shows also Customer-Info, CustomerInfoMaxSize max. size in characters of Customer-Info).
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::Overview'}->{'Preview'} = { 'CustomerInfo' => '0', 'CustomerInfoMaxSize' => '18', 'DefaultPreViewLines' => '18', 'DefaultViewNewLine' => '90', 'Module' => 'Kernel::Output::HTML::TicketOverviewPreview', 'Name' => 'Preview', 'NameShort' => 'L', 'StripEmptyLines' => '0', 'TicketActionsPerTicket' => '0' };</pre>

B.2.53. Frontend::Agent::ToolBarModule

B.2.53.1. Frontend::ToolBarModule###1-Ticket::TicketLocked

Description	Value
Description:	Agent interface notification module to check the used charset.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule

Description	Value
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::ToolBarModule'}->{'1-Ticket::TicketLocked'} = { 'CssClass' => 'Locked', 'CssClassNew' => 'Locked New', 'CssClassReached' => 'Locked Reached', 'Module' => 'Kernel::Output::HTML::ToolBarTicketLocked' };</pre>

B.2.53.2. Frontend::ToolBarModule###1-Ticket::TicketWatcher

Description	Value
Description:	Agent interface notification module to see the number of watched tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::ToolBarModule'}->{'1-Ticket::TicketWatcher'} = { 'CssClass' => 'Watcher', 'CssClassNew' => 'Watcher New', 'CssClassReached' => 'Watcher Reached', 'Module' => 'Kernel::Output::HTML::ToolBarTicketWatcher' };</pre>

B.2.53.3. Frontend::ToolBarModule###1-Ticket::TicketResponsible

Description	Value
Description:	Agent interface notification module to see the number of tickets an agent is responsible for.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::ToolBarModule'}->{'1-Ticket::TicketResponsible'} = 'CssClass' => 'Responsible', 'CssClassNew' => 'Responsible New', 'CssClassReached' => 'Responsible Reached', 'Module' => 'Kernel::Output::HTML::ToolBarTicketResponsible' }; </pre>

B.2.53.4. Frontend::ToolBarModule###410-Ticket::AgentTicketEmail

Description	Value
Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::ToolBarModule'}->{'410-Ticket::AgentTicketEmail'} = 'AccessKey' => '1', 'Action' => 'AgentTicketEmail', 'CssClass' => 'EmailTicket', 'Link' => 'Action=AgentTicketEmail', 'Module' => 'Kernel::Output::HTML::ToolBarLink', 'Name' => 'New email ticket', 'Priority' => '1009999' }; </pre>

B.2.53.5. Frontend::ToolBarModule###400-Ticket::AgentTicketPhone

Description	Value
Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::ToolBarModule'}->{'400-Ticket::AgentTicketPhone'} = 'AccessKey' => '1', 'Action' => 'AgentTicketPhone', 'CssClass' => 'PhoneTicket', 'Link' => 'Action=AgentTicketPhone', 'Module' => 'Kernel::Output::HTML::ToolBarLink', 'Name' => 'New phone ticket', 'Priority' => '1009998' }; </pre>

B.2.53.6. Frontend::ToolBarModule###410-Ticket::AgentTicketQueue

Description	Value
Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::ToolBarModule'}->{'410-Ticket::AgentTicketQueue'} = 'AccessKey' => 'q', 'Action' => 'AgentTicketQueue', 'CssClass' => 'QueueView', 'Link' => 'Action=AgentTicketQueue', 'Module' => 'Kernel::Output::HTML::ToolBarLink', 'Name' => 'Queue view', 'Priority' => '1012000' }; </pre>

B.2.53.7. Frontend::ToolBarModule###410-Ticket::AgentTicketStatus

Description	Value
Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::ToolBarModule'}->{'410-Ticket::AgentTicketStatus'} = 'AccessKey' => 'o', 'Action' => 'AgentTicketStatusView', 'CssClass' => 'StatusView', 'Link' => 'Action=AgentTicketStatusView', 'Module' => 'Kernel::Output::HTML::ToolBarLink', 'Name' => 'Status view', 'Priority' => '1012001' }; </pre>

B.2.53.8. Frontend::ToolBarModule###410-Ticket::AgentTicketEscalation

Description	Value
Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::ToolBarModule'}->{'410-Ticket::AgentTicketEscalation 'AccessKey' => 'w', 'Action' => 'AgentTicketEscalationView', 'CssClass' => 'EscalationView', 'Link' => 'Action=AgentTicketEscalationView', 'Module' => 'Kernel::Output::HTML::ToolBarLink', 'Name' => 'Escalation view', 'Priority' => '1012002' }; </pre>

B.2.53.9. Frontend::ToolBarModule###1-Ticket::TicketSearchProfile

Description	Value
Description:	Agent interface module to access search profiles via nav bar.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::ToolBarModule'}->{'1-Ticket::TicketSearchProfile'} = 'Block' => 'ToolBarSearchProfile', 'Description' => 'Search-Template', 'MaxWidth' => '40', 'Module' => 'Kernel::Output::HTML::ToolBarTicketSearchProfile', 'Name' => 'Search-Template' }; </pre>

B.2.53.10. Frontend::ToolBarModule###1-Ticket::TicketSearchFulltext

Description	Value
Description:	Agent interface module to access fulltext search via nav bar.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::ToolBarModule'}->{'1-Ticket::TicketSearchFulltext'} 'Block' => 'ToolBarSearchFulltext', 'CSS' => 'Core.Agent.Toolbar.FulltextSearch.css', 'Description' => 'Fulltext-Search', 'Module' => 'Kernel::Output::HTML::ToolBarTicketSearchFulltext', 'Name' => 'Fulltext-Search', 'Size' => '10' }; </pre>

B.2.54. Frontend::Customer

B.2.54.1. Ticket::Frontend::CustomerTicketOverviewSortable

Description	Value
Description:	Controls if customers have the ability to sort their tickets.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketOverviewSortable'} = '';</code>

B.2.54.2. Ticket::Frontend::CustomerTicketOverviewCustomEmptyText

Description	Value
Description:	Custom text for the page shown to customers that have no tickets yet.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketOverviewCustomEmptyText'} = { 'Button' => 'Create your first ticket', 'Text' => 'Please click the button below to create your first ticket.', 'Title' => 'Welcome!' };</code>

B.2.54.3. Frontend::CustomerUser::Item###9-OpenTickets

Description	Value
Description:	Customer item (icon) which shows the open tickets of this customer as info block.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Frontend::CustomerUser::Item'}->{'9-OpenTickets'} = { 'Action' => 'AgentTicketSearch', 'Attributes' => 'StateType=Open;', 'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css', 'CSSClassNoOpenTicket' => 'NoOpenTicket', 'CSSClassOpenTicket' => 'OpenTicket', 'Module' => 'Kernel::Output::HTML::CustomerUserGenericTicket', 'Subaction' => 'Search', 'Target' => '_blank', 'Text' => 'Open Tickets' };</pre>

B.2.54.4. CustomerFrontend::CommonObject###QueueObject

Description	Value
Description:	Path of the file that stores all the settings for the QueueObject object for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerFrontend::CommonObject'}->{'QueueObject'} = 'Kernel::</pre>

B.2.54.5. CustomerFrontend::CommonObject###TicketObject

Description	Value
Description:	Path of the file that stores all the settings for the TicketObject for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerFrontend::CommonObject'}->{'TicketObject'} = 'Kernel::</pre>

B.2.54.6. CustomerFrontend::CommonParam###Action

Description	Value
Description:	Defines the default used Frontend-Module if no Action parameter given in the url on the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerFrontend::CommonParam'}->{'Action'} = 'CustomerTicket'</code>

B.2.54.7. CustomerFrontend::CommonParam###TicketID

Description	Value
Description:	Default ticket ID used by the system in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerFrontend::CommonParam'}->{'TicketID'} = '';</code>

B.2.55. Frontend::Customer::ModuleMetaHead

B.2.55.1. CustomerFrontend::HeaderMetaModule###2-TicketSearch

Description	Value
Description:	Module to generate html OpenSearch profile for short ticket search in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleMetaHead
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'CustomerFrontend::HeaderMetaModule'}->{'2-TicketSearch'} = { 'Action' => 'CustomerTicketSearch', 'Module' => 'Kernel::Output::HTML::HeaderMetaTicketSearch' };</code>

B.2.56. Frontend::Customer::ModuleRegistration

B.2.56.1. CustomerFrontend::Module###CustomerTicketOverView

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerFrontend::Module'}->{'CustomerTicketOverView'} = { 'Description' => 'Overview of customer tickets', 'NavBar' => [{ 'AccessKey' => 'm', 'Block' => "", 'Description' => 'My Tickets', 'Link' => 'Action=CustomerTicketOverView;Subaction=MyTickets', 'LinkOption' => "", 'Name' => 'My Tickets', 'NavBar' => "", 'Prio' => '110', 'Type' => "" }, { 'AccessKey' => 'c', 'Block' => "", 'Description' => 'Company Tickets', 'Link' => 'Action=CustomerTicketOverView;Subaction=CompanyTickets', 'LinkOption' => "", 'Name' => 'Company Tickets', 'NavBar' => "", 'Prio' => '120', 'Type' => "" }], 'NavBarName' => 'Ticket', 'Title' => 'Overview' }; </pre>

B.2.56.2. CustomerFrontend::Module###CustomerTicketMessage

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

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

B.2.56.3. CustomerFrontend::Module###CustomerTicketZoom

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}->{'CustomerTicketZoom'} = { 'Description' => 'Ticket zoom view', 'Loader' => { 'JavaScript' => ['Core.Customer.TicketZoom.js', 'Core.UI.Popup.js'] }, 'NavBarName' => 'Ticket', 'Title' => 'Zoom' }; </pre>

B.2.56.4. CustomerFrontend::Module###CustomerTicketPrint

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}->{'CustomerTicketPrint'} = { 'Description' => 'Customer Ticket Print Module', 'NavBarName' => "", 'Title' => 'Print' }; </pre>

B.2.56.5. CustomerFrontend::Module###CustomerZoom

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}->{'CustomerZoom'} = { 'Description' => 'compat mod', 'NavBarName' => "", 'Title' => " "};</pre>

B.2.56.6. CustomerFrontend::Module###CustomerTicketAttachment

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}->{'CustomerTicketAttachment'} = { 'Description' => 'To download attachments', 'NavBarName' => "", 'Title' => " "};</pre>

B.2.56.7. CustomerFrontend::Module###CustomerTicketSearch

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

Description	Value
Config-Setting:	<pre> \$Self->{'CustomerFrontend::Module'}->{'CustomerTicketSearch'} = { 'Description' => 'Customer ticket search', 'NavBar' => [{ 'AccessKey' => 's', 'Block' => "", 'Description' => 'Search', 'Link' => 'Action=CustomerTicketSearch', 'LinkOption' => "", 'Name' => 'Search', 'NavBar' => "", 'Prio' => '300', 'Type' => "" }], 'NavBarName' => 'Ticket', 'Title' => 'Search' }; </pre>

B.2.57. Frontend::Customer::Preferences

B.2.57.1. CustomerPreferencesGroups###ShownTickets

Description	Value
Description:	Defines all the parameters for the ShownTickets object in the customer preferences of the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerPreferencesGroups'}->{'ShownTickets'} = { 'Active' => '1', 'Column' => 'User Profile', 'Data' => { '15' => '15', '20' => '20', '25' => '25', '30' => '30' }, 'DataSelected' => '25', 'Key' => 'Max. displayed tickets', 'Label' => 'Number of displayed tickets', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserShowTickets', 'Prio' => '4000' }; </pre>

B.2.57.2. CustomerPreferencesGroups###RefreshTime

Description	Value
Description:	Defines all the parameters for the RefreshTime object in the customer preferences of the customer interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerPreferencesGroups'}->{'RefreshTime'} = { 'Active' => '1', 'Column' => 'User Profile', 'Data' => { " => 'off', '10' => '10 minutes', '15' => '15 minutes', '2' => ' 2 minutes', '5' => ' 5 minutes', '7' => ' 7 minutes' }, 'DataSelected' => "", 'Key' => 'Refresh interval', 'Label' => 'Ticket overview', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserRefreshTime', 'Prio' => '4000' }; </pre>

B.2.58. Frontend::Customer::Ticket::ViewNew

B.2.58.1.

Ticket::Frontend::CustomerTicketMessage###NextScreenAfterNewTicket

Description	Value
Description:	Determines the next screen after new customer ticket in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'NextScreenAfterNe </pre>

B.2.58.2. Ticket::Frontend::CustomerTicketMessage###Priority

Description	Value
Description:	Allows customers to set the ticket priority in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'Priority'} = '1'</code>

B.2.58.3. Ticket::Frontend::CustomerTicketMessage###PriorityDefault

Description	Value
Description:	Defines the default priority of new customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'PriorityDefault'}</code>

B.2.58.4. Ticket::Frontend::CustomerTicketMessage###Queue

Description	Value
Description:	Allows customers to set the ticket queue in the customer interface. If this is set to 'No', QueueDefault should be configured.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'Queue'} = '1';</code>

B.2.58.5. Ticket::Frontend::CustomerTicketMessage###QueueDefault

Description	Value
Description:	Defines the default queue for new customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	0
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'QueueDefault'} =</code>

B.2.58.6. Ticket::Frontend::CustomerTicketMessage###Service

Description	Value
Description:	Allows customers to set the ticket service in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'Service'} = '1';</code>

B.2.58.7. Ticket::Frontend::CustomerTicketMessage###SLA

Description	Value
Description:	Allows customers to set the ticket SLA in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'SLA'} = '1';</code>

B.2.58.8. Ticket::Frontend::CustomerTicketMessage###StateDefault

Description	Value
Description:	Defines the default state of new customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'StateDefault'} =</code>

B.2.58.9. Ticket::Frontend::CustomerTicketMessage###ArticleType

Description	Value
Description:	Defines the default type for article in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'ArticleType'} =</code>

B.2.58.10. Ticket::Frontend::CustomerTicketMessage###SenderType

Description	Value
Description:	Sender type for new tickets from the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'SenderType'} =</code>

B.2.58.11. Ticket::Frontend::CustomerTicketMessage###HistoryType

Description	Value
Description:	Defines the default history type in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'HistoryType'} =</code>

B.2.58.12. Ticket::Frontend::CustomerTicketMessage###HistoryComment

Description	Value
Description:	Comment for new history entries in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'HistoryComment'}</code>

B.2.58.13. CustomerPanelSelectionType

Description	Value
Description:	Defines the receipt target of the tickets ("Queue" shows all queues, "SystemAddress" displays all system addresses) in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelSelectionType'} = 'Queue';</code>

B.2.58.14. CustomerPanelSelectionString

Description	Value
Description:	Determines the strings that will be shown as receipt (To:) of the ticket in the customer interface. For Queue as CustomerPanelSelectionType, "<Queue>" shows the names of the queues, and for SystemAddress, "<Realname> <<Email>>" shows the name and email of the receipt.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	0

Description	Value
Config-Setting:	<code>\$Self->{'CustomerPanelSelectionString'} = '<Queue>';</code>

B.2.58.15. CustomerPanelOwnSelection

Description	Value
Description:	Determines which queues will be valid for ticket's recipients in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'CustomerPanelOwnSelection'} = { 'Junk' => 'First Queue', 'Misc' => 'Second Queue' };</code>

B.2.58.16. CustomerPanel::NewTicketQueueSelectionModule

Description	Value
Description:	Module for To-selection in new ticket screen in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanel::NewTicketQueueSelectionModule'} =</code>

'Kernel::Out

B.2.58.17. Ticket::Frontend::CustomerTicketMessage###TicketFreeText

Description	Value
Description:	Ticket free text options shown in the ticket message screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required. NOTE. If you want to display these fields also in the ticket zoom of the customer interface, you have to enable them in CustomerTicketZoom###AttributesView.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'TicketFreeText'} '1' => '0', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0', '15' => '0', '16' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' };</pre>

B.2.58.18. Ticket::Frontend::CustomerTicketMessage###TicketFreeTime

Description	Value
Description:	Ticket free time options shown in the ticket message screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required. NOTE. If you want to display these fields also in the ticket zoom of the customer interface, you have to enable them in CustomerTicketZoom###AttributesView.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'TicketFreeTime'} '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

B.2.58.19. Ticket::Frontend::CustomerTicketMessage###ArticleFreeText

Description	Value
Description:	Article free text options shown in the ticket message screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required. NOTE. If you want to display these fields also in the ticket zoom of the customer interface, you have to enable them in CustomerTicketZoom###AttributesView.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'ArticleFreeText'} '1' => '0', '2' => '0', '3' => '0' };</pre>

B.2.59. Frontend::Customer::Ticket::ViewSearch

B.2.59.1. Ticket::CustomerTicketSearch::SearchLimit

Description	Value
Description:	Maximum number of tickets to be displayed in the result of a search in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::CustomerTicketSearch::SearchLimit'} = '5000';</pre>

B.2.59.2. Ticket::CustomerTicketSearch::SearchPageShown

Description	Value
Description:	Number of tickets to be displayed in each page of a search result in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::CustomerTicketSearch::SearchPageShown'} = '40';</pre>

B.2.59.3. Ticket::CustomerTicketSearch::SortBy::Default

Description	Value
Description:	Defines the default ticket attribute for ticket sorting in a ticket search of the customer interface.

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

B.2.59.4. Ticket::CustomerTicketSearch::Order::Default

Description	Value
Description:	Defines the default ticket order of a search result in the customer interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::CustomerTicketSearch::Order::Default'} = 'Down';</code>

B.2.59.5.

Ticket::Frontend::CustomerTicketSearch###ExtendedSearchCondition

Description	Value
Description:	Allows extended search conditions in ticket search of the customer interface. With this feature you can search w. g. with this kind of conditions like "(key1&&key2)" or "(key1 key2)".
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'ExtendedSearchCond</code>

B.2.59.6. Ticket::Frontend::CustomerTicketSearch###TicketFreeText

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

Description	Value
Description:	Ticket free text options shown in the ticket search screen in the customer interface. Possible settings: 0 = Disabled and 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'TicketFreeText'} = '1' => '0', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0', '15' => '0', '16' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' };</pre>

B.2.59.7. Ticket::Frontend::CustomerTicketSearch###TicketFreeTime

Description	Value
Description:	Ticket free time options shown in the ticket search screen in the customer interface. Possible settings: 0 = Disabled and 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'TicketFreeTime'} = '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

B.2.59.8. Ticket::Frontend::CustomerTicketSearch###SearchCSVData

Description	Value
Description:	Data used to export the search result in CSV format.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchCSVData'} = 'TicketNumber', 'Age', 'Created', 'Closed', 'State', 'Priority', 'Lock', 'CustomerID', 'CustomerName', 'From', 'Subject'];</pre>

B.2.60. Frontend::Customer::Ticket::ViewZoom

B.2.60.1.

Ticket::Frontend::CustomerTicketZoom###NextScreenAfterFollowUp

Description	Value
Description:	Determines the next screen after the follow up screen of a zoomed ticket in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'NextScreenAfterFollowUp'} =</pre>

B.2.60.2. Ticket::Frontend::CustomerTicketZoom###ArticleType

Description	Value
Description:	Defines the default type of the note in the ticket zoom screen of the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'ArticleType'} = 'we'</pre>

B.2.60.3. Ticket::Frontend::CustomerTicketZoom###SenderType

Description	Value
Description:	Defines the default sender type for tickets in the ticket zoom screen of the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'SenderType'} = 'cus</code>

B.2.60.4. Ticket::Frontend::CustomerTicketZoom###HistoryType

Description	Value
Description:	Defines the history type for the ticket zoom action, which gets used for ticket history in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'HistoryType'} = 'Fo</code>

B.2.60.5. Ticket::Frontend::CustomerTicketZoom###HistoryComment

Description	Value
Description:	Defines the history comment for the ticket zoom action, which gets used for ticket history in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'HistoryComment'} =</code>

B.2.60.6. Ticket::Frontend::CustomerTicketZoom###Priority

Description	Value
Description:	Allows customers to change the ticket priority in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'Priority'} = '1';</code>

B.2.60.7. Ticket::Frontend::CustomerTicketZoom###PriorityDefault

Description	Value
Description:	Defines the default priority of follow up customer tickets in the ticket zoom screen in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'PriorityDefault'} =</code>

B.2.60.8. Ticket::Frontend::CustomerTicketZoom###State

Description	Value
Description:	Allows choosing the next compose state for customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'State'} = '1';</code>

B.2.60.9. Ticket::Frontend::CustomerTicketZoom###StateDefault

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

Description	Value
Description:	Defines the default next state for a ticket after customer follow up in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'StateDefault'} = 'o</code>

B.2.60.10. Ticket::Frontend::CustomerTicketZoom###StateType

Description	Value
Description:	Defines the next possible states for customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'StateType'} = ['open', 'closed'];</code>

B.2.60.11. Ticket::Frontend::CustomerTicketZoom###AttributesView

Description	Value
Description:	Shows the activated ticket attributes in the customer interface (0 = Disabled and 1 = Enabled).
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'AttributesView'} = 'ArticleFreeText1' => '0', 'ArticleFreeText2' => '0', 'ArticleFreeText3' => '0', 'Priority' => '1', 'Queue' => '1', 'SLA' => '0', 'Service' => '0', 'State' => '1', 'TicketFreeText1' => '0', 'TicketFreeText10' => '0', 'TicketFreeText11' => '0', 'TicketFreeText12' => '0', 'TicketFreeText13' => '0', 'TicketFreeText14' => '0', 'TicketFreeText15' => '0', 'TicketFreeText16' => '0', 'TicketFreeText2' => '0', 'TicketFreeText3' => '0', 'TicketFreeText4' => '0', 'TicketFreeText5' => '0', 'TicketFreeText6' => '0', 'TicketFreeText7' => '0', 'TicketFreeText8' => '0', 'TicketFreeText9' => '0', 'TicketFreeTime1' => '0', 'TicketFreeTime2' => '0', 'TicketFreeTime3' => '0', 'TicketFreeTime4' => '0', 'TicketFreeTime5' => '0', 'TicketFreeTime6' => '0', 'Type' => '0' }; </pre>

B.2.61. Frontend::Queue::Preferences

B.2.61.1. QueuePreferences###Comment2

Description	Value
Description:	Parameters of the example queue attribute Comment2.
Group:	Ticket
SubGroup:	Frontend::Queue::Preferences
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'QueuePreferences'}->{'Comment2'} = { 'Block' => 'TextArea', 'Cols' => '50', 'Desc' => 'Define the queue comment 2.', 'Label' => 'Comment2', 'Module' => 'Kernel::Output::HTML::QueuePreferencesGeneric', 'PrefKey' => 'Comment2', 'Rows' => '5' }; </pre>

B.2.62. Frontend::SLA::Preferences

B.2.62.1. SLAPreferences###Comment2

Description	Value
Description:	Parameters of the example SLA attribute Comment2.
Group:	Ticket
SubGroup:	Frontend::SLA::Preferences
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'SLAPreferences'}->{'Comment2'} = { 'Block' => 'TextArea', 'Cols' => '50', 'Desc' => 'Define the sla comment 2.', 'Label' => 'Comment2', 'Module' => 'Kernel::Output::HTML::SLAPreferencesGeneric', 'PrefKey' => 'Comment2', 'Rows' => '5' };</pre>

B.2.63. Frontend::Service::Preferences

B.2.63.1. ServicePreferences###Comment2

Description	Value
Description:	Parameters of the example service attribute Comment2.
Group:	Ticket
SubGroup:	Frontend::Service::Preferences
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'ServicePreferences'}->{'Comment2'} = { 'Block' => 'TextArea', 'Cols' => '50', 'Desc' => 'Define the service comment 2.', 'Label' => 'Comment2', 'Module' => 'Kernel::Output::HTML::ServicePreferencesGeneric', 'PrefKey' => 'Comment2', 'Rows' => '5' };</pre>

Appendix C. Credits

Like other open source projects we want to thank some people for their help and support. The following list is surely incomplete and we've definitely forgotten someone; sorry for that! Just drop us a note if you are not on this list.

The following persons have especially pushed the project or are still active supporters:

- Robert Kehl, who has created the Win32-installer for the 1.x releases. Thanks a lot, Robert!
- Torsten Werner, who maintains the Debian-installer for OTRS. Thanks a lot, Torsten, by making OTRS also available for the Debian community.
- Nils Jeppe (mirror Hamburg, Germany), Bryan Fullerton (mirror Toronto, Canada), Eberhard Mönkeberg (mirror Göttingen, Germany), Timo Dreger (mirror Düsseldorf, Germany) and Netmonic (mirror Vienna, Austria), who are mirroring our ftp server. Thanks a lot, with your help it is always possible to download OTRS quickly!
- Anja Schneider, who is helping with the translation of this manual and with the correction and revision of the German texts. Many thanks for your help and patience, Anja!
- We receive many ideas, via the OTRS mailing lists, for system improvements, patches or bugfixes and great support for all users is available. Thanks a lot to all people on the mailing lists for your active assistance!

Also we want to give a big thank you to the following persons:

- Martin Scherbaum
- Carsten Gross
- Harald Müller
- Stefan Schmidt
- Milisav Radmanic
- Uli Hecht
- Norman Walsh
- Heiko Baumann
- Atif Ghaffar
- Pablo Ruiz Garcia
- Dan Rau
- Christoph Kaulich

- Mark Jackson
- Diane Shieh
- Bernard Choppy
- Carl Bailey
- Phil Davis
- Edwin D. Vinas
- Lars Müller
- Vladimir Gerdjikov
- Fred van Dijk
- Sebastien Guilbaud
- Wiktor Wodecki
- Arnold Ligtoet
- Antti Kämäräinen
- Nicolas Goralski
- Gilberto Cezar de Almeida
- Jorge Becerra
- Eddie Urenda
- Stella Power
- Andreas Haase
- Reiner Keller
- Covert Jake
- Moshe Leibovitch
- Björn Jacke
- Remo Catelotti
- Alfons Obermeyer
- Michael Rech
- Danie Theron
- Richard Paradies
- Art Powell

Appendix D. GNU Free Documentation License

Version 1.1, March 2000

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

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. 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 non-commercially, 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.