

OTRS 2.4 - Admin Manual

OTRS 2.4 - Admin Manual

Copyright © 2003-2009 OTRS AG

René Bakker, Hauke Böttcher, Stefan Bedorf, Shawn Beasley, Jens Bothe, Udo Bretz, Martin Edenhofer, Manuel Hecht, Christopher Kuhn, André Mindermann, Henning Oschwald, Thomas Raith, Stefan Rother, Burchard Steinbild

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 and Windows Vista 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	xi
1. Basics about trouble ticket systems	1
What is a trouble ticket system and why do you need it?	1
What is a trouble ticket?	2
2. OTRS - Open Ticket Request System	3
Basics	3
Features	3
New features of OTRS 2.4	5
New features of OTRS 2.3	8
New features of OTRS 2.2	9
New features of OTRS 2.1	10
Hard and software requirements	11
Community	12
Commercial Support and Services for OTRS	13
3. Installation / Upgrade of the OTRS framework	14
The simple way - Installation of pre-built packages	14
Installing the rpm on a SUSE Linux server	14
Installing OTRS on a Debian system	19
Installing OTRS on Microsoft Windows systems	19
Installation from source (Linux, Unix)	19
Preparing the installation from source	19
Installation of Perl modules	20
Configuring the apache web server	23
Configuring the database	25
Setting up the cron jobs for OTRS	29
Upgrading the OTRS Frameworks	33
Upgrading .tar.gz	33
Upgrading RPM	34
4. First steps in OTRS	36
Agent web interface	36
Customer web interface	36
Public web interface	37
First login	38
An overview to the web interface	39
What is a queue?	41
User preferences	42
5. The admin area of OTRS	44
Basics	44
Users, groups and roles	44
Users	44
Groups	45
Roles	48
Customer users and customer groups	50
Customer users	50
Customer groups	50
Queues	51
Salutations, signatures, attachments and responses	53
Salutations	54
Signatures	55
Attachments	56
Auto answers	58

Email addresses	59
Notifications	60
SMIME	62
PGP	63
States	63
SysConfig	64
Using mail accounts	65
Filtering incoming messages	66
Executing automated jobs with the GenericAgent	68
Admin email	70
Session management	70
System Log	71
SQL queries via the SQL box	72
Package manager	73
6. Configuring the system	75
The config files of OTRS	75
Configuring the system through the web interface	75
7. Sending/Receiving emails	77
Sending emails	77
Via Sendmail (default)	77
Via SMTP server or smarthost	77
Receiving emails	77
Mail accounts configured via the OTRS GUI	77
Via command line program and e.g. procmail (PostMaster.pl)	78
Fetching emails via POP3 or IMAP and fetchmail for PostMaster.pl	79
Filtering/dispatching by OTRS/PostMaster modules (for more complex dispatching)	79
8. Time related functions	81
Setting up business hours, holidays and time zones	81
Business Hours	81
Fixed date holidays	81
TimeVacationDaysOneTime	81
Automated Unlocking	82
9. Ticket responsibility and ticket watching	83
Ticket responsibility	83
Ticket watching	84
10. Customize the PDF output	87
11. Using external backends	88
Customer data	88
Customer user backend	88
Database (Default)	88
LDAP	92
Use more than one customer backend with OTRS	95
Backends to authenticate agents and customer users	98
Authentication backends for agents	98
Authentication backends for customer users	101
Customize the customer self registration	103
Customizing the web interface	103
Customer mapping	104
Customize the customer_user table in the OTRS DB	105
12. States	107
Predefined states	107
New	107
Open	107

Pending reminder	107
Pending auto close-	107
Pending auto close+	107
Merged	107
Closed Successful	107
Closed Unsuccessful	107
Customizing states	108
13. Modifying ticket priorities	111
14. Creating your own themes	112
15. Localization of the OTRS frontend	113
16. PGP	114
17. S/MIME	118
18. Access Control Lists (ACLs)	122
19. Stats module	126
Handling of the module by the agent	126
Overview	127
Generate and view stats	129
Edit - New	132
Import	138
Administration of the stat module by the OTRS administrator	139
Permission settings, groups and queues	139
SysConfig	140
Administration of the stats module by the system administrator	140
Data base table	140
List of all files	140
Caching	141
mkStats.pl	141
Automated stat generation - Cron	141
Static stats	141
Using old static stats	142
Default stats	142
20. Additional applications	143
Calendar	143
Content manager	143
File manager	143
Web mailer	144
FAQ	144
System status	145
21. Performance Tuning	146
OTRS	146
TicketIndexModule	146
TicketStorageModule	146
Database	147
MySQL	147
PostgreSQL	147
Webserver	147
Pre-establish database connections	147
Preloaded modules - startup.pl	147
Reload Perl modules when updated on disk	148
Choosing the Right Strategy	148
mod_gzip/mod_deflate	148
22. Backing up the system	149
Backup	149
Restore	149

A. Additional Resources	151
Homepage OTRS.org	151
Mailing lists	151
Bug tracking	152
Commercial Support	153
B. Configuration Options Reference	154
Framework	154
Core	154
Core::LinkObject	163
Core::Log	165
Core::MIME-Viewer	167
Core::MirrorDB	168
Core::PDF	169
Core::Package	172
Core::PerformanceLog	174
Core::SOAP	175
Core::Sendmail	176
Core::Session	178
Core::SpellChecker	183
Core::Stats	184
Core::Stats::Graph	186
Core::Time	191
Core::Time::Calendar1	195
Core::Time::Calendar2	199
Core::Time::Calendar3	202
Core::Time::Calendar4	205
Core::Time::Calendar5	208
Core::Time::Calendar6	212
Core::Time::Calendar7	215
Core::Time::Calendar8	218
Core::Time::Calendar9	221
Core::Web	225
Core::WebUserAgent	233
Crypt::PGP	234
Crypt::SMIME	236
Frontend::Admin::ModuleRegistration	237
Frontend::Agent	250
Frontend::Agent::Dashboard	254
Frontend::Agent::ModuleMetaHead	256
Frontend::Agent::ModuleNotify	257
Frontend::Agent::ModuleRegistration	259
Frontend::Agent::NavBarModule	265
Frontend::Agent::Preferences	266
Frontend::Agent::Stats	270
Frontend::Customer	274
Frontend::Customer::Auth	282
Frontend::Customer::ModuleMetaHead	290
Frontend::Customer::ModuleNotify	291
Frontend::Customer::ModuleRegistration	292
Frontend::Customer::Preferences	294
Frontend::Public	298
Frontend::Public::ModuleRegistration	298
Ticket	299
Core::ArticleFreeText	299

Core::FulltextSearch	303
Core::PostMaster	304
Core::Ticket	316
Core::TicketACL	331
Core::TicketBulkAction	332
Core::TicketFreeText	333
Core::TicketFreeTextDefault	358
Core::TicketFreeTime	365
Core::TicketWatcher	372
Frontend::Admin::ModuleRegistration	373
Frontend::Agent	383
Frontend::Agent::CustomerSearch	391
Frontend::Agent::Dashboard	393
Frontend::Agent::ModuleNotify	397
Frontend::Agent::ModuleRegistration	398
Frontend::Agent::NavBarModule	412
Frontend::Agent::Preferences	414
Frontend::Agent::Ticket::ArticleAttachmentModule	418
Frontend::Agent::Ticket::ArticleComposeModule	419
Frontend::Agent::Ticket::ArticleViewModule	420
Frontend::Agent::Ticket::ArticleViewModulePre	421
Frontend::Agent::Ticket::MenuModule	421
Frontend::Agent::Ticket::MenuModulePre	429
Frontend::Agent::Ticket::ViewBounce	432
Frontend::Agent::Ticket::ViewBulk	434
Frontend::Agent::Ticket::ViewClose	437
Frontend::Agent::Ticket::ViewCompose	444
Frontend::Agent::Ticket::ViewCustomer	449
Frontend::Agent::Ticket::ViewEmailNew	449
Frontend::Agent::Ticket::ViewEscalation	453
Frontend::Agent::Ticket::ViewForward	454
Frontend::Agent::Ticket::ViewFreeText	457
Frontend::Agent::Ticket::ViewHistory	465
Frontend::Agent::Ticket::ViewMailbox	465
Frontend::Agent::Ticket::ViewMerge	467
Frontend::Agent::Ticket::ViewMove	468
Frontend::Agent::Ticket::ViewNote	471
Frontend::Agent::Ticket::ViewOwner	478
Frontend::Agent::Ticket::ViewPending	486
Frontend::Agent::Ticket::ViewPhoneNew	494
Frontend::Agent::Ticket::ViewPhoneOutbound	498
Frontend::Agent::Ticket::ViewPriority	503
Frontend::Agent::Ticket::ViewQueue	511
Frontend::Agent::Ticket::ViewResponsible	514
Frontend::Agent::Ticket::ViewSearch	522
Frontend::Agent::Ticket::ViewStatus	526
Frontend::Agent::Ticket::ViewZoom	527
Frontend::Agent::TicketOverview	530
Frontend::Customer	532
Frontend::Customer::ModuleRegistration	533
Frontend::Customer::Preferences	537
Frontend::Customer::Ticket::ViewNew	539
Frontend::Customer::Ticket::ViewSearch	543
Frontend::Customer::Ticket::ViewZoom	546

Frontend::Queue::Preferences	549
Frontend::SLA::Preferences	549
Frontend::Service::Preferences	550
C. Credits	551
D. GNU Free Documentation License	553
0. PREAMBLE	553
1. APPLICABILITY AND DEFINITIONS	553
2. VERBATIM COPYING	554
3. COPYING IN QUANTITY	554
4. MODIFICATIONS	555
5. COMBINING DOCUMENTS	556
6. COLLECTIONS OF DOCUMENTS	556
7. AGGREGATION WITH INDEPENDENT WORKS	557
8. TRANSLATION	557
9. TERMINATION	557
10. FUTURE REVISIONS OF THIS LICENSE	557
How to use this License for your documents	558

List of Tables

3.1. Needed Perl modules for OTRS	20
3.2. Description of the several cron job scripts	30
5.1. Default groups after OTRS has been installed	46
5.2. Rights in the user groups of OTRS	47
5.3. Events for auto answers	58
5.4. Function of the different X-OTRS-headers	67
A.1. Mailinglists	151

List of Examples

5.1. Sort spam mails into a specific queue	68
7.1. .fetchmailrc	79
7.2. Example jobs for the filter module Kernel::System::PostMaster::Filter::Match	80
7.3. Example job for the filter module Kernel::System::PostMaster::Filter::CMD	80
11.1. Configuring a DB customer backend	88
11.2. Using company tickets with a DB backend	92
11.3. Configuring a LDAP customer backend	92
11.4. Using Company tickets with a LDAP backend	94
11.5. Using more than one customer backend with OTRS	95
11.6. Authenticate agents against a DB backend	98
11.7. Authenticate agents against a LDAP backend	99
11.8. Authenticate agents using HTTPBasic	100
11.9. Authenticate agents against a radius backend	101
11.10. Customer user authentication against a DB backend	101
11.11. Customer user authentication against a LDAP backend	101
11.12. Customer user authentication with HTTPBasic	102
11.13. Customer user authentication against a radius backend	103
18.1. ACL which only allows to move tickets with ticket priority 5 into a queue	122
18.2. ACL, which disables the closing of tickets in the raw queue and hides the close button	122
18.3. ACL, which removes the status for all agents, and only provides it for a group	123
19.1. Definition of a value series - one element	137
19.2. Definition of a value series - two elements	137

Preface

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

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

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

Chapter 1. Basics about trouble ticket systems

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

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.

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 informations about the hardware and software requirements for OTRS. Additionally, this chapter tells you how to get commercial support for OTRS, should you require it, and how to contact the community.

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.

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.

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 WYSIWIG 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.

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.

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.

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.

Hard 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 server as well as a working Perl environment with some additional modules. The web server and Perl has to be installed on the same machine as OTRS. The database back-end can be installed on the local or on an other host.

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

For database back-ends, you can use MySQL, PostgreSQL, Oracle, MSSQL or DB2. If you use MySQL you have the advantage that you can configure your database and some system settings during the installation through a web front-end.

For Perl, we recommend using at least version 5.8.8. 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

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

Databases

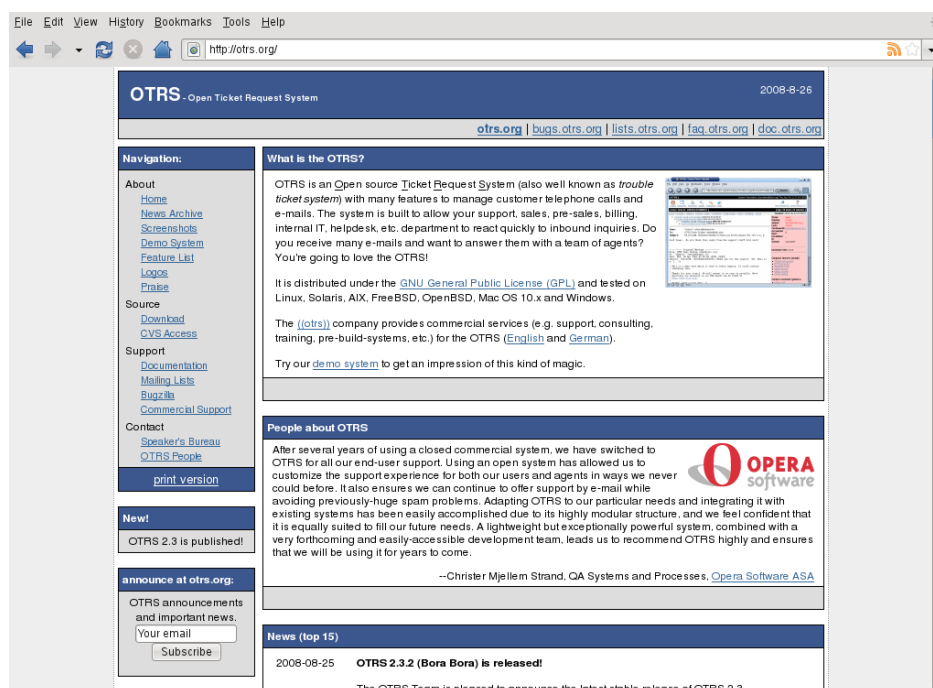
- MySQL 4.1 or higher
- PostgreSQL 8.0 or higher
- Oracle 10g or higher
- DB2 8 or higher
- MSSQL 2000 or higher

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

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

Community

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



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

Commercial Support and Services for OTRS

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

OTRS Group provides subscription support services, customization, consulting and training for *OTRS Help Desk* [<http://www.otrs.com/en/products/help-desk/>] and *OTRS ITSM* [<http://www.otrs.com/en/products/itsm/>]. It also provides *Best Practice Product Editions* [<http://www.otrs.com/solutions/>]. With these Editions, OTRS Group helps organizations to design, deploy and optimize OTRS for each unique environment. Additionally, OTRS Group provides hosted versions including *OTRS OnDemand* [<http://www.otrs.com/en/solutions/ondemand/>] and *Managed OTRS* [<http://www.otrs.com/en/solutions/managed-otrs/>].

You can find more detailed information about OTRS Group on <http://www.otrs.com> [<http://www.otrs.com>] and you can contact us via email on [sales at otrs.com](mailto: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 RPM or with the Windows Installer.

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

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

The simple way - Installation of pre-built packages

The simplest and most comfortable way to install OTRS is to use pre-built packages. Many pre-built packages of OTRS can be found in the download area on <http://www.otrs.org> [<http://www.otrs.org>]. The following sections describe the installation of OTRS with a 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.

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 the SUSE versions from 7.x to 10.0. Before you start the installation please have a look on <http://www.otrs.org> [<http://www.otrs.org/>] and check, if a newer OTRS rpm package is available. Please use the newer rpm package, if available.

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

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

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

Next steps:

```
[SuSEconfig]
Execute 'SuSEconfig' to configure the 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:~ #

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

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

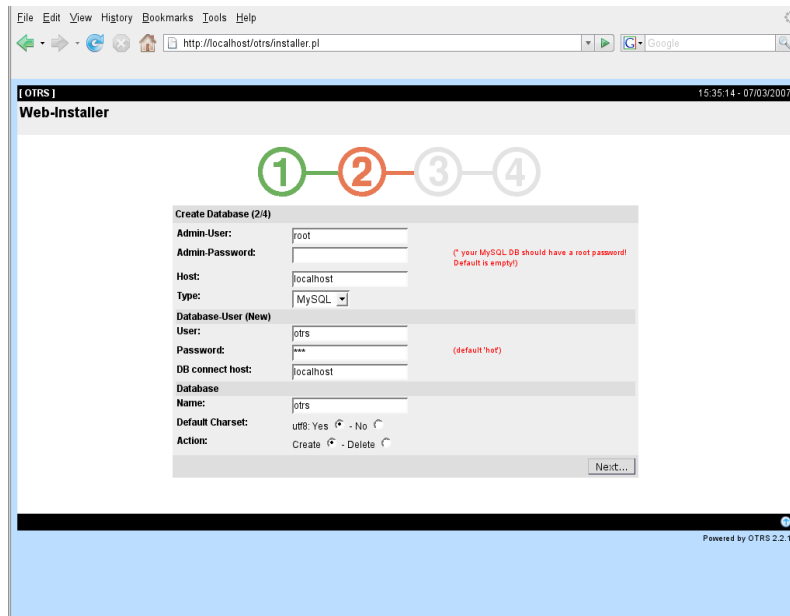
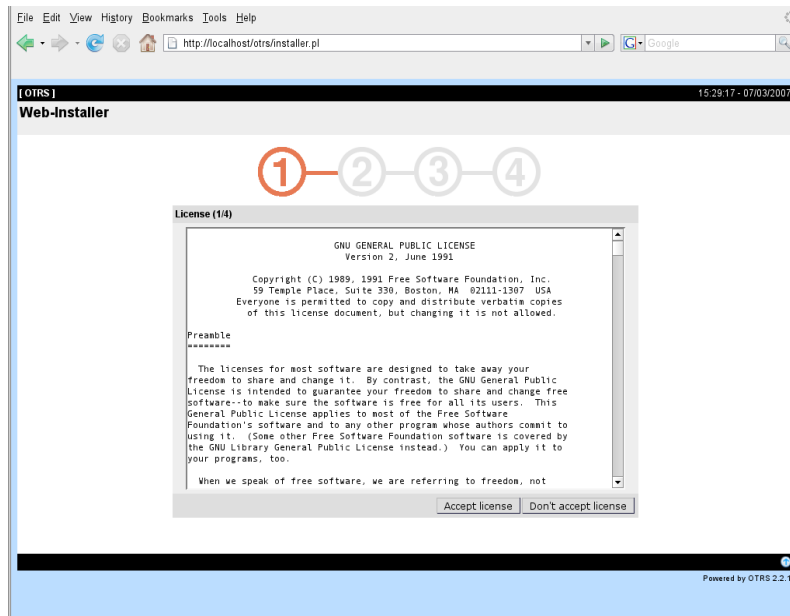
The installation of the OTRS rpm is finished. Restart your web server to load the OTRS specific changes in your web server configuration:

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

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

<http://localhost/otrs/installer.pl> [<http://localhost/otrs/installer.pl>]

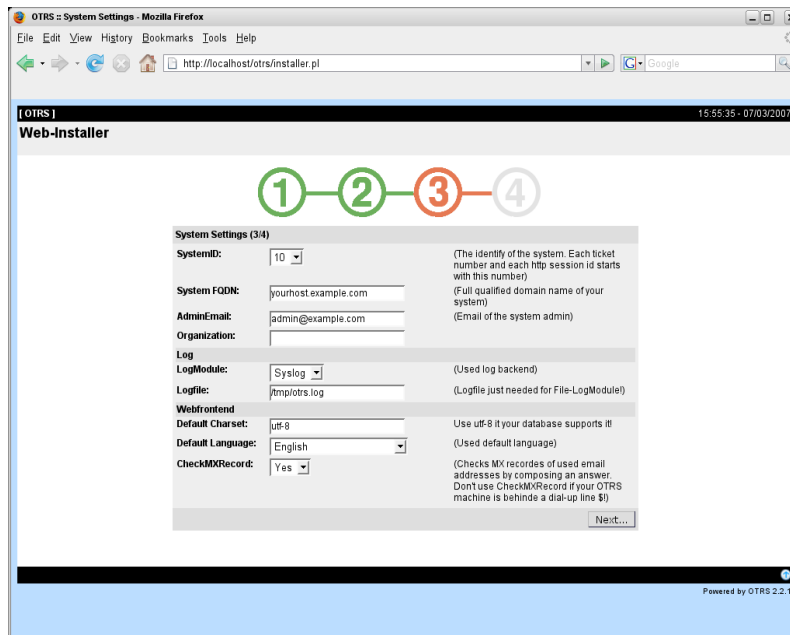
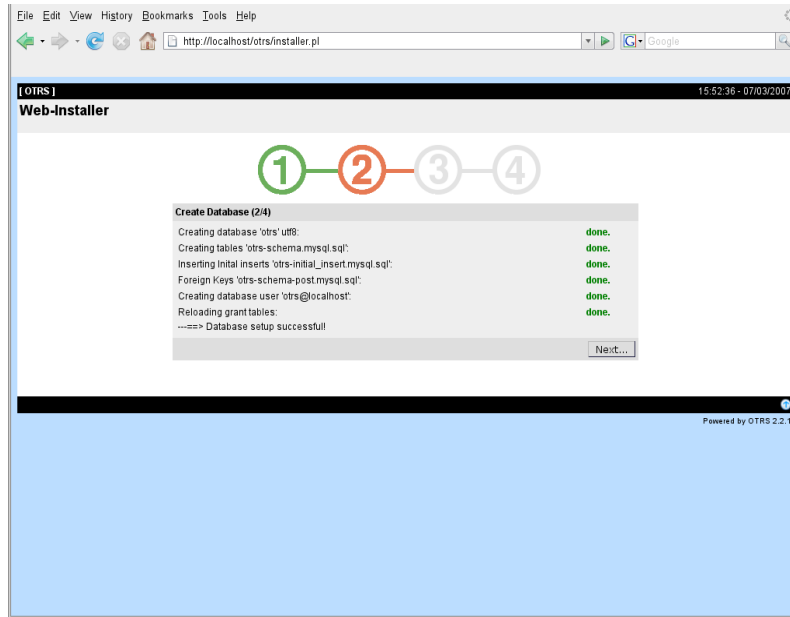
The web installer starts. Please follow the steps and setup the system.



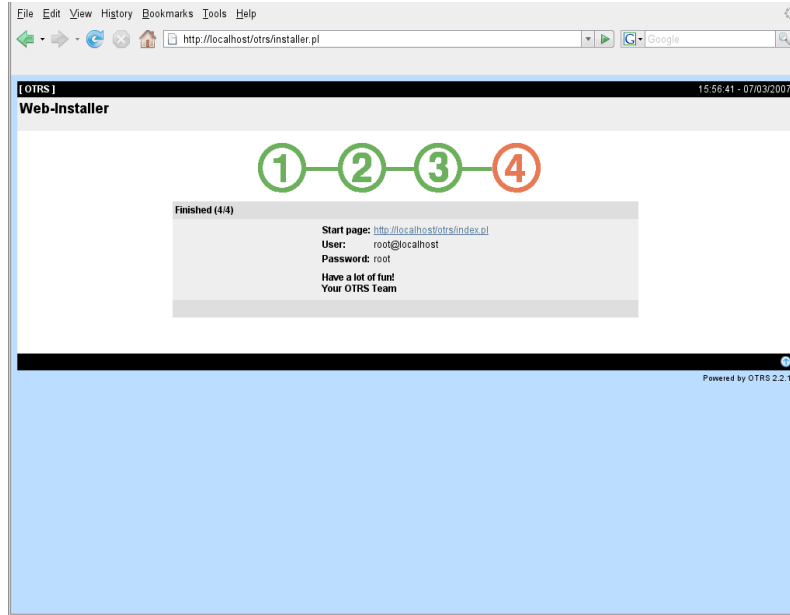
Warning

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

Installation / Upgrade of the OTRS framework



Installation / Upgrade of the OTRS framework



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

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

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

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

done

done
linux:~ #
```

Now the installation of OTRS is finished and you should be able to work with the system. To log into the web interface of OTRS, use the address `http://localhost/otrs/index.pl` [`http://localhost/otrs/index.pl`] in your web browser. Log in as OTRS administrator and configure the system for

your needs. To log in as OTRS administrator use the username `root@localhost` and the default password `root`.

Warning

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

Installing OTRS on a Debian system

Torsten Werner, the maintainer of the Debian OTRS package, kindly provided excellent documentation on the installation of OTRS on Debian systems. The document can be found on: http://www.writely.com/View?docid=drm3kmx_0cbr3x9 [http://docs.google.com/View?docid=drm3kmx_0cbr3x9].

Installing OTRS on Microsoft Windows systems

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

Important

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

Installation from source (Linux, Unix)

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/download/> [<http://www.otrs.org/download/>]

Unpack the archive for example with `tar` into the directory `/opt` and rename the directory from `otrs-2.4.x` to `otrs`:

```
linux:/opt# tar xf /tmp/otrs-2.4.x.tar.gz
linux:/opt# mv otrs-2.4.-x otrs
linux:/opt# ls
otrs
linux:/opt#
```

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:

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

```
linux:/opt# usermod -G nogroup otrs
linux:/opt#
```

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`.

```
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
```

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

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

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

Installation of Perl modules

OTRS needs some additional Perl modules. If you install OTRS from source, you'll have to install these modules manually. This can either be done with the 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.

OTRS requires the following additional Perl modules:

Table 3.1. Needed Perl modules for OTRS

Name	Description
CGI	This module is needed by the web interface of OTRS.
Date::Pcalc	This module is needed for date calculations. OTRS uses this module for example in time specific calculations for tickets.
DBI	OTRS needs this module to connect to the database back-end.
DBD::mysql	Module with special functions to connect to the MySQL database back-end. (only required if you use MySQL)
DBD::pg	Module with special functions to connect to the PostgreSQL database back-end. (only required if you use PostgreSQL)

Name	Description
Digest::MD5	This module makes it possible to use the md5 algorithm.
LWP::UserAgent	Module to process HTTP requests.
MIME::Base64	En- and decoding Base64 strings, e.g. for mail attachments.
MIME::Tools	This module provide some tools to process messages with MIME parts.
Mail::Internet	Used for OTRS email processing.
Net::DNS	Perl interface to the domain name system.
Net::POP3	This module contains procedures to access and process messages on a POP3 server.
Net::POP3Client	This module contains procedures to access and process messages on a POP3 server.
IO::Socket::SSL	Required for SSL connections to your mail server.
Net::LDAP	Perl interface to a LDAP directory. You only need to install this module, if you want to use a LDAP back-end.
Net::SMTP	Module that contains procedures to send emails.
Authen::SASL	SASL authentication framework, e.g. needed for the authentication against mail servers.
Net::SMTP::SSL	Required if you use SSL connections to your SMTP server.
GD	Interface to the GD graphics library. You only need to install this module, if you want to use the stats module in OTRS.
GD::Text, GD::Graph, GD::Graph::lines, GD::Text::Align	Some more text- and graphic tools for the GD graphics library. You only need to install this modules, if you want to use the stats module of OTRS.
SOAP::Lite	This module is needed if you'd like to use the XML-RPC interface of OTRS.
XML::Parser	This module is needed to read and write xml configuration files. The graphical configuration front-end of OTRS uses this module.
PDF::API2, Compress::Zlib	This module are needed to generate the PDF output for reports, search results or for the ticket print view.

You can verify which modules you need to install with **otrs.checkModules**. The script is located in the `bin` directory in the home directory of the OTRS user.

```
linux:~# cd /opt/otrs/bin/  
linux:/opt/otrs/bin# ./otrs.checkModules
```



```
o CGI.....ok (v3.43)
o Date::Pcalc.....ok (v1.2)
o Date::Format.....ok (v2.22)
o DBI.....ok (v1.607)
o DBD::mysql.....ok (v4.008)
o Digest::MD5.....ok (v2.36_01)
o Crypt::PasswdMD5.....ok (v1.3)
o LWP::UserAgent.....ok (v5.819)
o Encode::HanExtra.....ok (v0.23)
o IO::Scalar.....ok (v2.110)
o IO::Wrap.....ok (v2.110)
o MIME::Base64.....ok (v3.07_01)
o Mail::Internet.....ok (v2.04)
o MIME::Tools.....ok (v5.427)
o Net::DNS.....ok (v0.63)
o Net::POP3.....ok (v2.29)
o Mail::POP3Client.....ok (v2.18 )
  o IO::Socket::SSL.....ok (v1.18)
o Net::IMAP::Simple.....ok (v1.17)
  o Net::IMAP::Simple::SSL.....ok (v1.3)
o Net::SMTP.....ok (v2.31)
  o Authen::SASL.....ok (v2.12)
  o Net::SMTP::SSL.....ok (v1.01)
o Net::LDAP.....ok (v0.39)
o GD.....ok (v2.39)
  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 PDF::API2.....ok (v0.73)
o SOAP::Lite.....ok (v0.710.08)
o XML::Parser.....ok (v2.36)
linux:/opt/otrs/bin#
```

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 lets you know, if other modules are needed.

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

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

Configuring the apache web server

This section describes the basic configuration of the apache web server for OTRS for use with `mod_cgi`. 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 to a file called `otrs.conf`.

First find the example configuration file in `otrs/scripts/` that is fitting for your version of the apache web server. Either: `Apache2 apache2-httpd-new.include.conf`

```
# --
# added for OTRS (http://otrs.org/)
# $Id: installation-and-basic-configuration.xml,v 1.30.2.1 2011/10/24
# 08:54:25 mg Exp $
# --

# agent, admin and customer frontend
ScriptAlias /otrs/ "/opt/otrs/bin/cgi-bin/"
Alias /otrs-web/ "/opt/otrs/var/httpd/htdocs/"

# if mod_perl is used
<IfModule mod_perl.c>

    # load all otrs modules
    PerlRequire /opt/otrs/scripts/apache2-perl-startup.pl

    # Apache::Reload - Reload Perl Modules when Changed on Disk
    PerlModule Apache2::Reload
    PerlInitHandler Apache2::Reload
    PerlModule Apache2::RequestRec

    # set mod_perl2 options
    <Location /otrs>
#       ErrorDocument 403 /otrs/customer.pl
       ErrorDocument 403 /otrs/index.pl
       SetHandler perl-script
       PerlResponseHandler ModPerl::Registry
       Options +ExecCGI
       PerlOptions +ParseHeaders
       PerlOptions +SetupEnv
       Order allow,deny
       Allow from all
    </Location>

</IfModule>
```

```
# directory settings
<Directory "/opt/otrs/bin/cgi-bin/">
    AllowOverride None
    Options +ExecCGI -Includes
    Order allow,deny
    Allow from all
</Directory>
<Directory "/opt/otrs/var/httpd/htdocs/">
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

# MaxRequestsPerChild (so no apache child will be to big!)
MaxRequestsPerChild 400
```

Apache1 apache-httpd.include.conf

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

# agent, admin and customer frontend (mod_alias required!)
ScriptAlias /otrs/ "/opt/otrs/bin/cgi-bin/"
Alias /otrs-web/ "/opt/otrs/var/httpd/htdocs/"

# if mod_perl is used
<IfModule mod_perl.c>

    # load all otrs modules (speed improvement!)
    # Perlrequire /opt/otrs/scripts/apache-perl-startup.pl

    # Apache::StatINC - Reload %INC files when updated on disk
    # (just use it for testing, setup, ... not for high-load systems)
    # PerlInitHandler Apache::StatINC

    <Location /otrs>
#       ErrorDocument 403 /otrs/customer.pl
        ErrorDocument 403 /otrs/index.pl
        SetHandler perl-script
        PerlHandler Apache::Registry
        Options ExecCGI
        PerlSendHeader On
        PerlSetupEnv On
    </Location>

</IfModule>

# MaxRequestsPerChild (so no apache child will be to big!)
MaxRequestsPerChild 400
```

Restart your web server to load the new configuration settings for the web server. On most systems you can start/restart your web server with the command **/etc/init.d/apache restart** or **/etc/init.d/apache2 restart**.

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

Now your 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 directory for `mod_perl.so` i.e.

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

to see if the module is already loaded somewhere, or not.

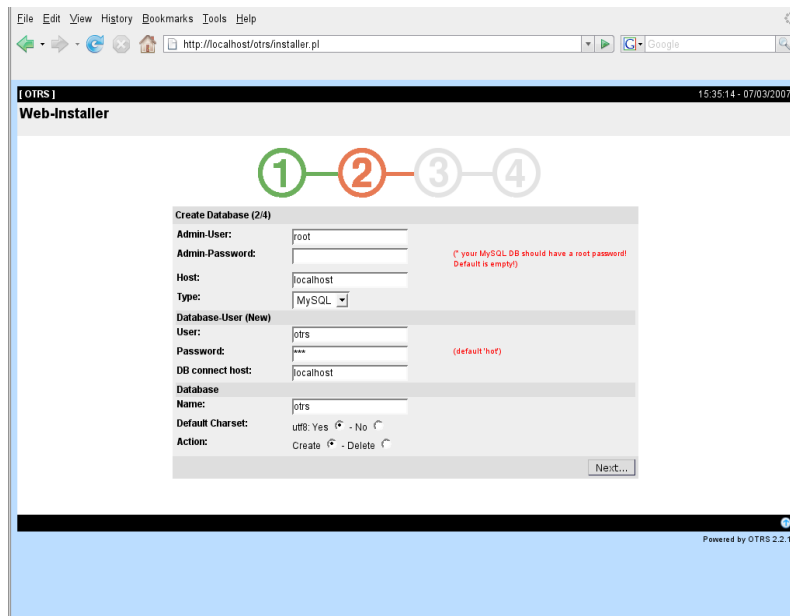
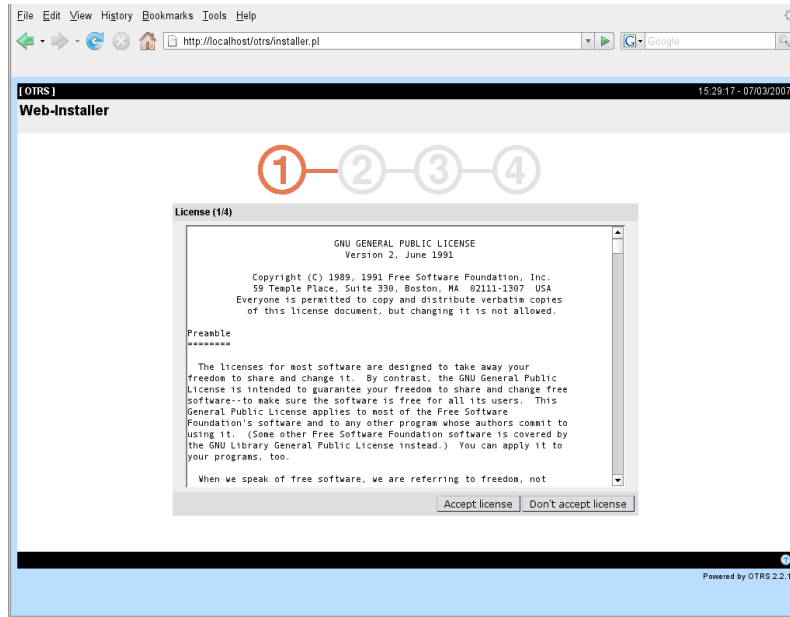
When you use the appropriate start script (listed above), and the module is loaded, the script (when commented in) `/opt/otrs/scripts/apache-perl-startup.pl` can be used to load the perl modules into memory one time, saving on load times and increasing performance.

Configuring the database

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

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

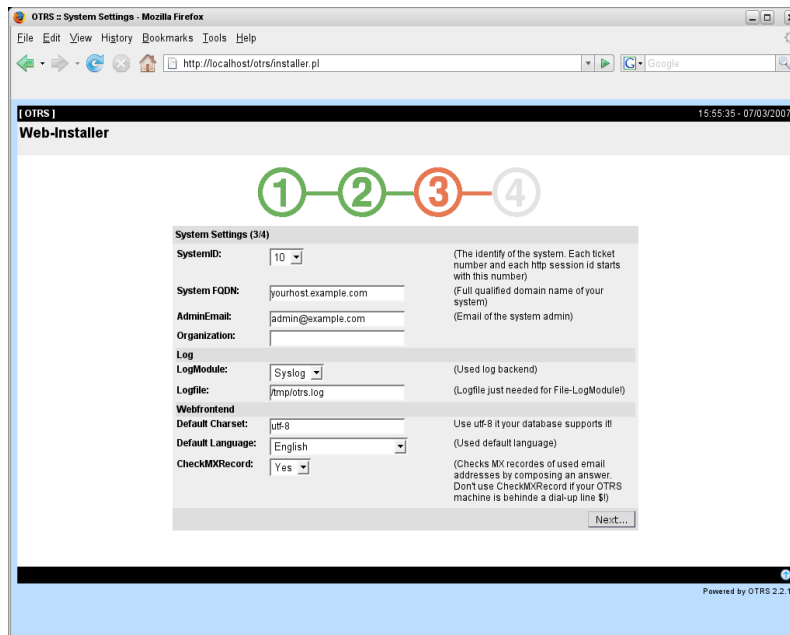
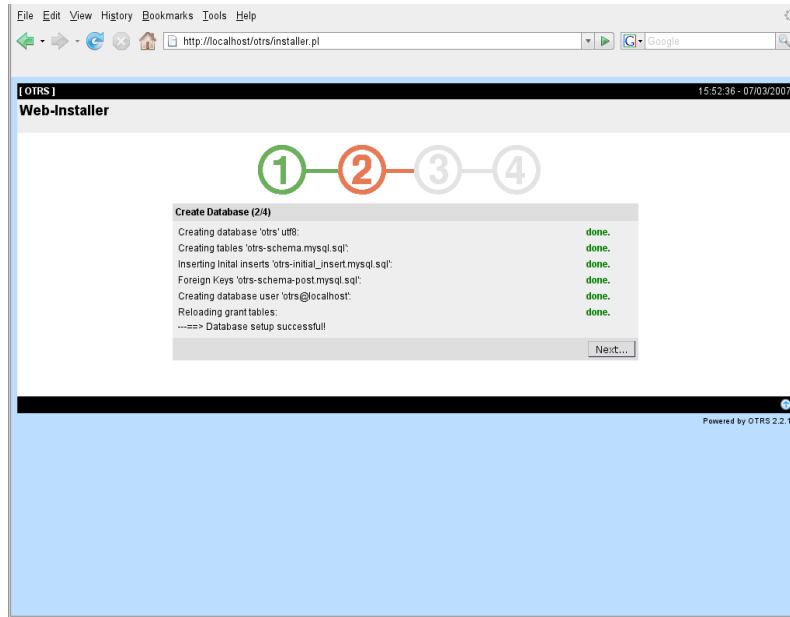
Installation / Upgrade of the OTRS framework

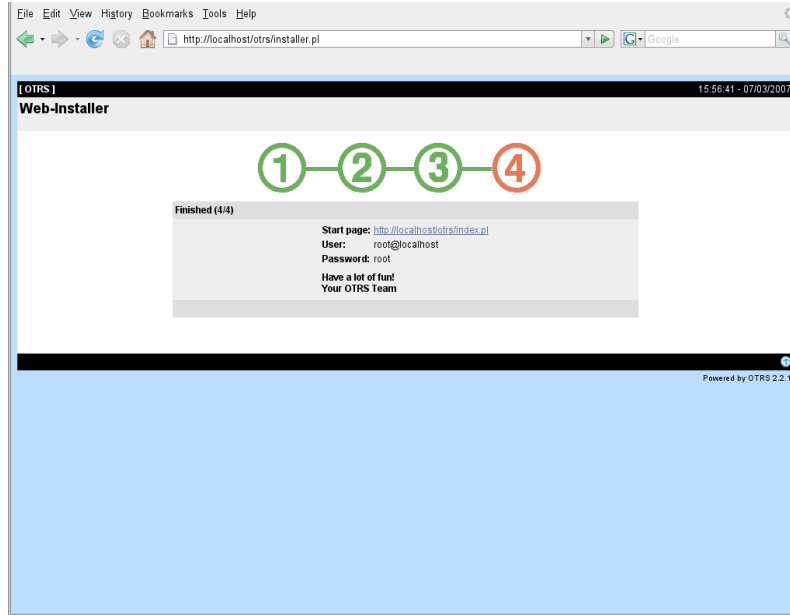


Warning

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

Installation / Upgrade of the OTRS framework





Installing the OTRS database manually

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

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

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

Create the OTRS database manually step by step

1. Creating the DB: Create the database, that you want to use for OTRS, with your database client or your database interface.
2. Creating the tables: With the `otrs-schema.DatabaseType.sql` files (e.g.. `otrs-schema.oracle.sql`, `otrs-schema.postgresql.sql`) you can create the tables in your OTRS database.
3. Inserting the initial system data: OTRS needs some initial system data to work properly (e.g. the different ticket states, ticket and notification types). Depending on

the type of your database Use 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` files to create these references (e.g. `otrs-schema-oracle.post.sql`, `otrs-schema-post.postgresql.sql`).

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

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

```
# 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';
```

Setting up the cron jobs for OTRS

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

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

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


All scripts are ending in .dist. You should copy them to files with no ending. If you are using bash, you might use the command listed 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.dist
  rebuild_ticket_index
aaa_base.dist                          pending_jobs
rebuild_ticket_index.dist
fetchmail                             pending_jobs.dist    session
fetchmail.dist                       postmaster           session.dist
generic_agent                        postmaster.dist      unlock
generic_agent-database              postmaster_pop3      unlock.dist
generic_agent-database.dist         postmaster_pop3.dist
linux:/opt/otrs/var/cron#
```

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

Table 3.2. Description of the several cron job scripts

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

To setup all cron jobs the script `bin/Cron.sh` can be used, which is located in the home directory of the OTRS user. the script needs a parameter when it is executed that tells if you like to install, remove or reinstall the cron jobs. The following parameters can be used:

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

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

Warning

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

```
linux:/opt/otrs/var/cron# cd /opt/otrs/bin/
linux:/opt/otrs/bin# su otrs
linux:~/bin$ ./Cron.sh start
/opt/otrs/bin
Cron.sh - start/stop OTRS cronjobs - <$Revision: 1.30.2.1 $>
Copyright (C) 2001-2009 OTRS AG, http://otrs.org/
(using /opt/otrs) done
linux:~/bin$ exit
exit
linux:/opt/otrs/bin#
```

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

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

# --
# cron/fetchmail - fetchmail cron of the OTRS
# Copyright (C) 2001-2011 OTRS AG, http://otrs.org/
# --
# $Id: installation-and-basic-configuration.xml,v 1.30.2.1 2011/10/24
# 08:54:25 mg Exp $
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# fetch every 5 minutes emails via fetchmail
#*/5 * * * * /usr/bin/fetchmail -a >> /dev/null

# --
# cron/generic_agent - GenericAgent.pl cron of the OTRS
# Copyright (C) 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/GenericAgent.pl -c
# "Kernel::Config::GenericAgentMove" >> /dev/null
# --
# 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/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/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 /etc/rc.d/init.d/otrs && /etc/rc.d/init.d/otrs
# cleanup >> /dev/null
10 0 * * * $HOME/bin/otrs.cleanup >> /dev/null
# --
# cron/postmaster_pop3 - postmaster_pop3 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/PostMasterPOP3.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/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/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/UnlockTickets.pl --timeout >> /dev/null

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

Upgrading the OTRS Frameworks

These instructions are for people upgrading OTRS from "2.3" to "2.4".

If you are running a lower version of OTRS you need 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 ...).

Upgrading .tar.gz

This section shows you how to upgrade OTRS from "2.3" to "2.4" with .tar.gz.

- Stop all your 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
```

```
shell> /etc/init.d/mysql stop
```

- 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.
- Make sure that you have backed up everything.
- Try this install on a separate machine, on a test machine first.
- Install the new release tar.gz

Note: The OTRS themes between 2.3 and 2.4 are not compatible (don't use the old themes)!

- Execute \$OTRS_HOME/bin/SetPermissions.pl!
- Update the database changes with (part 1/2)

```
MySQL: cat $OTRS_HOME/scripts/DBUpdate-to-2.4.mysql.sql | mysql -p -f -u root otrs
```

PostgreSQL: `cat $OTRS_HOME/scripts/DBUpdate-to-2.4.postgresql.sql | psql otrs`

- 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. Run `$OTRS_HOME/scripts/DBUpdate-to-2.4.pl`

- Update the database changes with (part 2/2):

MySQL: `cat $OTRS_HOME/scripts/DBUpdate-to-2.4-post.mysql.sql | mysql -p -f -u root otrs`

PostgreSQL: `cat $OTRS_HOME/scripts/DBUpdate-to-2.4-post.postgresql.sql | psql otrs`

- Restart your services

e. g. (depends on used services):

```
shell> /etc/init.d/mysql stop
```

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

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

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

- Check installed packages

In the package manager, check if all packages are still marked as correctly installed or if any require reinstallation or even a package upgrade.

Upgrading RPM

This section shows you how to upgrade OTRS from "2.3" to "2.4" with `.rpm`.

- Stop all your 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
```

```
shell> /etc/init.d/mysql stop
```

- 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.

- Make sure that you have backed up everything.
- Try this install on a separate machine, on a test machine first.
- Install the new release RPM

```
shell> rpm -Uvh otrs-2.4.x-noarch.rpm
```

Note: The OTRS themes between 2.3 and 2.4 are not compatible (don't use the old themes)!

- Update the database changes with (part 1)

MySQL: `cat $OTRS_HOME/scripts/DBUpdate-to-2.4.mysql.sql | mysql -p -f -u root otrs`

PostgreSQL: `cat $OTRS_HOME/scripts/DBUpdate-to-2.4.postgresql.sql | psql otrs`

- 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. Run `$OTRS_HOME/scripts/DBUpdate-to-2.4.pl`!
- Update the database changes with (part 2):

MySQL: `cat $OTRS_HOME/scripts/DBUpdate-to-2.4-post.mysql.sql | mysql -p -f -u root otrs`

PostgreSQL: `cat $OTRS_HOME/scripts/DBUpdate-to-2.4-post.postgresql.sql | psql otrs`

- Restart your services

e. g. (depends on used services):

```
shell> /etc/init.d/mysql stop
```

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

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

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

- Check installed packages

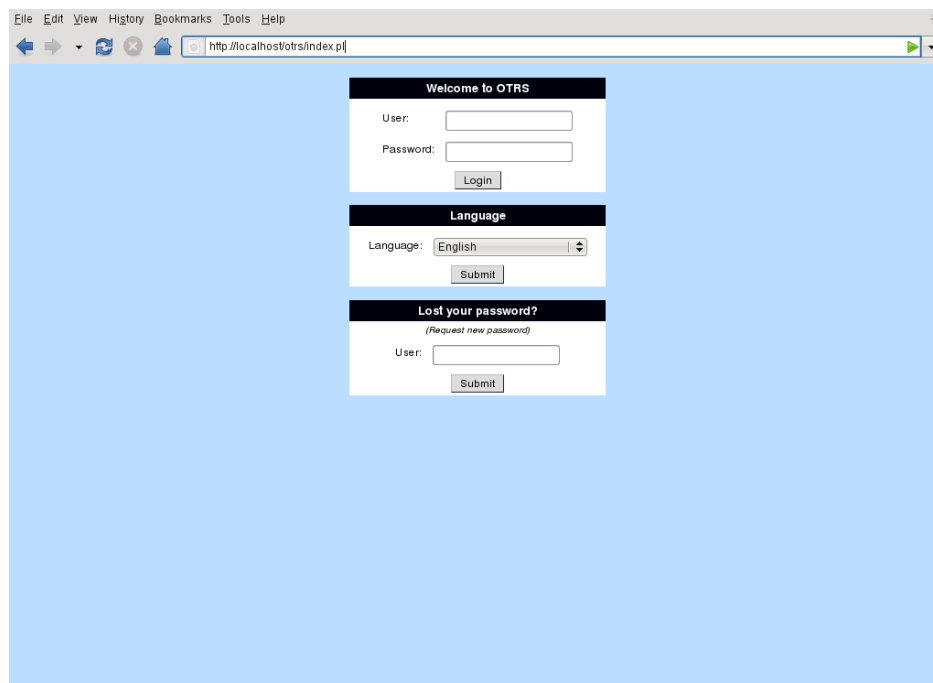
In the package manager, check if all packages are still marked as correctly installed or if any require reinstallation or even a package upgrade.

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.

Agent web interface

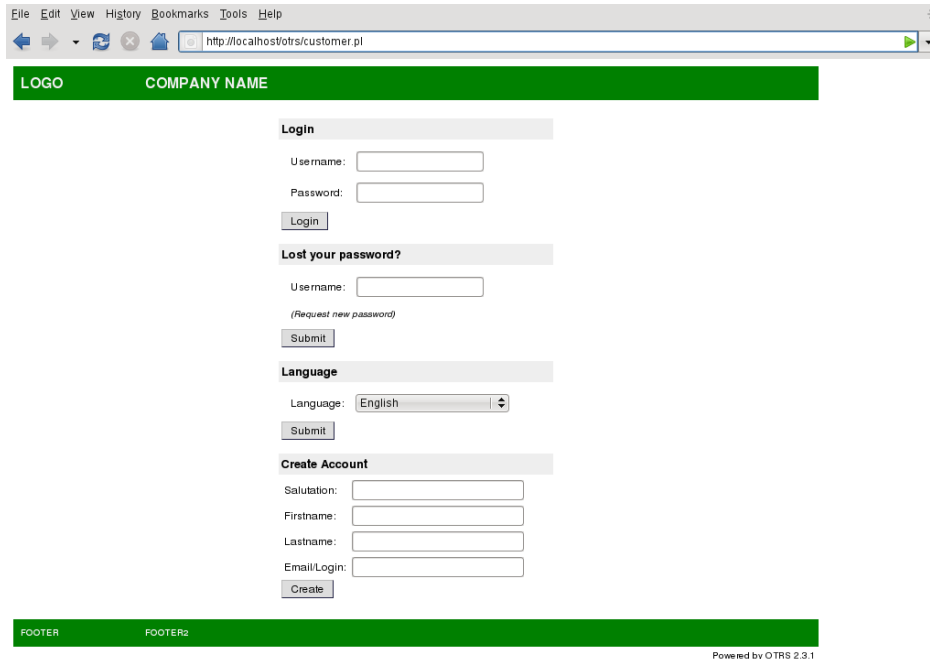
All agents use the agent web interface to work with OTRS. Agents answer the customer requests, create new tickets for customers or other agents, write tickets about telephone calls with customers, write FAQ entries or edit customer data, e.g.



Lets say that your OTRS host is reachable via the URL <http://www.example.com> [<http://www.example.com/>] then the agents and also the OTRS administrator can reach the login screen by using the address <http://www.example.com/otrs/index.pl> [<http://www.example.com/otrs/index.pl>] in a web browser.

Customer web interface

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

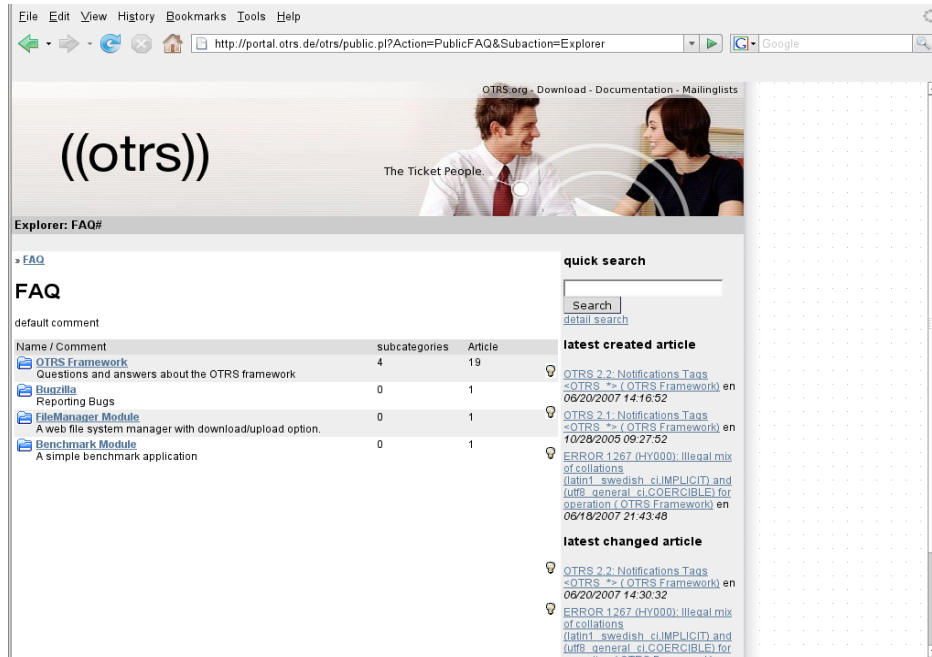


The screenshot shows a web browser window with the address bar displaying `http://localhost/otrs/customer.pl`. The page has a green header bar with "LOGO" and "COMPANY NAME" placeholders. The main content area contains four sections: "Login" with fields for "Username:" and "Password:" and a "Login" button; "Lost your password?" with a "Username:" field, a "(Request new password)" link, and a "Submit" button; "Language" with a "Language:" dropdown menu set to "English" and a "Submit" button; and "Create Account" with fields for "Salutation:", "Firstname:", "Lastname:", and "Email/Login:", along with a "Create" button. The footer is a green bar with "FOOTER" and "FOOTER2" placeholders, and a small text "Powered by OTRS 2.3.1" on the right.

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

Public web interface

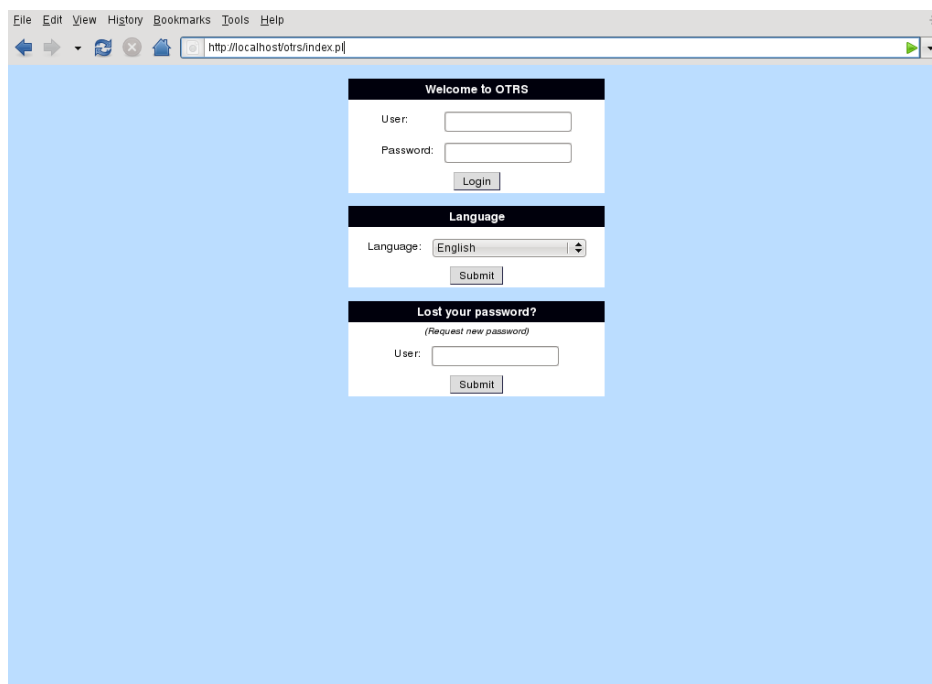
In addition to the web interfaces for agents and customers OTRS has a public web interface that is available through the FAQ-Module (must be installed separately). It provides public access to the FAQ system, and lets visitors search through the FAQ entries without any authorization.



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.

First login

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



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

Warning

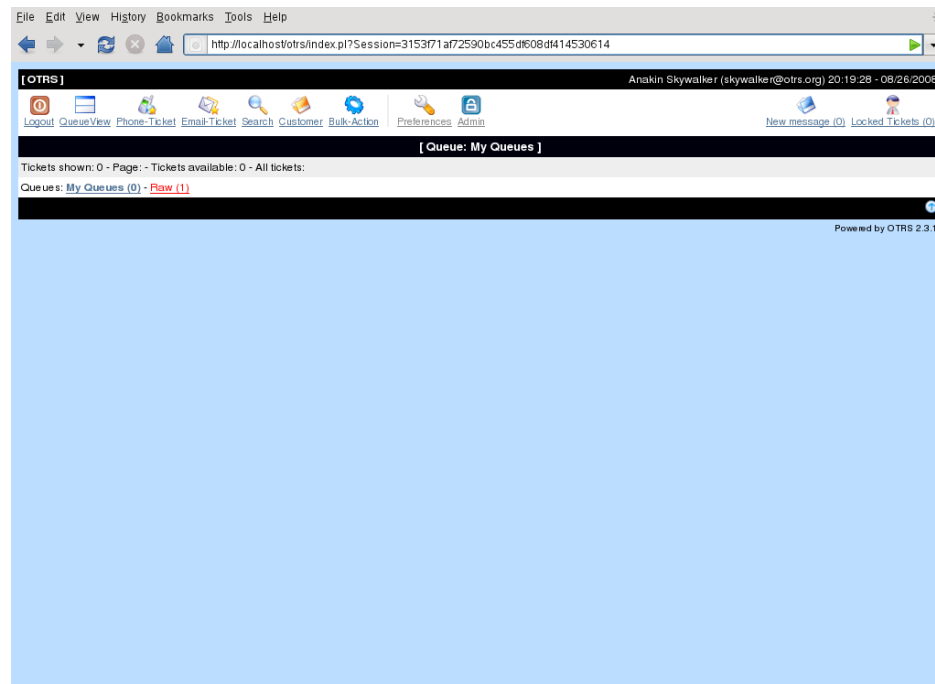
This account data are default on every new installed OTRS system. For that reason you should change the password for the OTRS administrator as fast as possible! This can be done via the preferences for the OTRS administrator account.

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

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

An overview to the web interface

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



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

The white bar below is the navigation bar. It shows buttons that enables you to navigate to different areas or modules of the system and lets you execute some global actions. The navbar

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

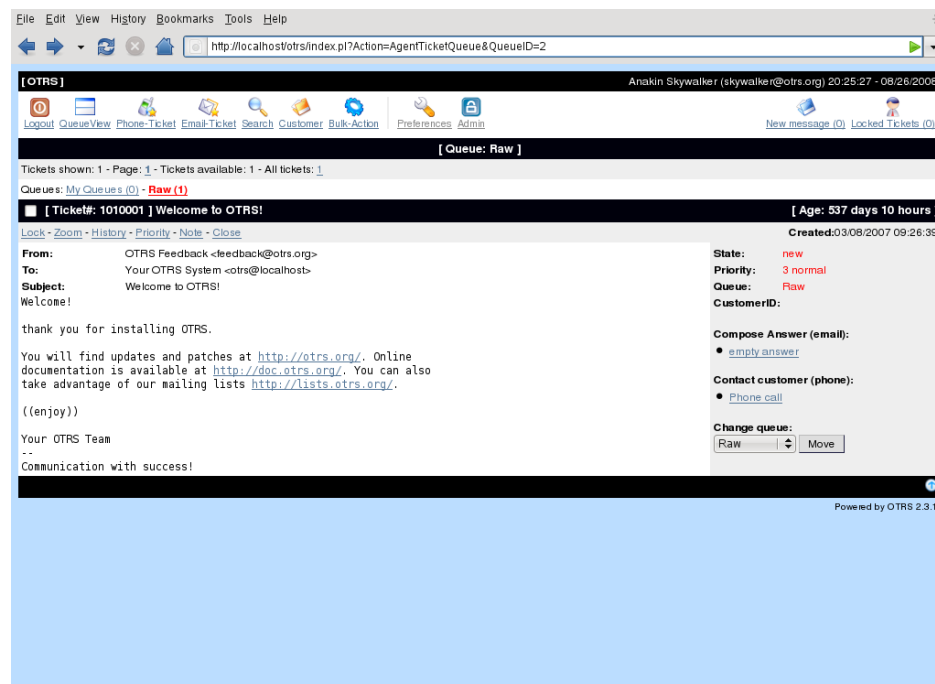
The middle part of the navbar shows the buttons for the navigation to the other modules of the system. After a standard installation you only see the "Admin" button, if you are logged in as OTRS administrator. If some additional application are installed, e.g. the file manager or the web mailer, buttons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

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

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

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

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



If you select the "raw" queue, the interface will be reloaded and the content of the queue is displayed. Now the lower part of the screen contains additional information about the new ticket in the "raw" queue. A black bar is filled with the ID of the ticket and its age. At the left site of this

bar the checkbox is available that lets you mark the ticket for the bulk action. The next grey line contains the actions that can be done with the ticket (e.g. Lock, Zoom, History). On the right side of this line the date and time when the ticket was created is displayed.

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

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

What is a queue?

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

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

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

To explain it more practically, remember Max's company described in the section about an example of a ticket system. Max has installed OTRS to get rid of his support chaos. He and his agents are using the system to manage the support requests for the video recorders. One queue that contains all requests is enough for this situation. But some time later Max also sells a DVD recorder. Now the customer not only has questions about the video recorder but also for the new product. More and more emails get into the one queue of Max's OTRS and it's hard to keep the overview. Some day Max decides to optimize the structure of his support system and adds two new queues, so now three queues are used by the system. Into the old queue (called "raw") all mails are stored if they arrive at the ticket system. The two new queues are one for the video recorder (called "video recorder") and the other one for the dvd recorder requests (called "dvd recorder"). Max tells Sandra to watch the "raw" queue and sort (dispatch) the mails either into "video recorder" or "dvd recorder", depending if a customer asks for support for a video recorder or dvd recorder. John only has access to the "video recorder" queue, Joe can only answer tickets in the "dvd recorder" queue. Max is able to edit tickets in all queues. Because OTRS supports access management for users, groups and roles, it is easy to setup queues that only are accessible for some specific accounts. Max could also use another way to get his requests into the different queues, with filter rules or if two different mail addresses are used Sandra only has to dispatch emails into the two other queues, that can't be dispatched automatically.

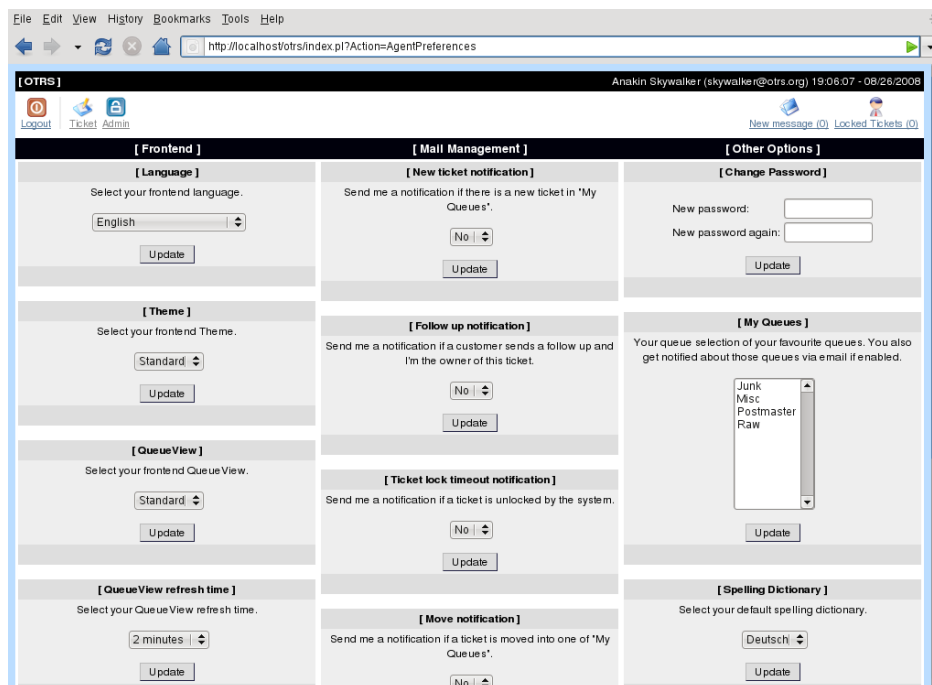
Sorting your incoming messages into different queues helps you to keep the support system structured and tidy. Because your agents are arranged into different groups with different access rights onto queues, the system can be optimized even more. Queues can be used to define work flow processes or they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

another queue called "sales". this queue could contain the sub queues "requests", "offers", "orders", "billing", e.g. Such a queue structure could help Max to optimize his order transactions. The better a system is structured, the less time is needed for the different tasks. That results in less working time that is needed for the different tasks and that saves money. Queues can help to optimize the processes in your company.

User preferences

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

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



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

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

Chapter 5. The admin area of OTRS

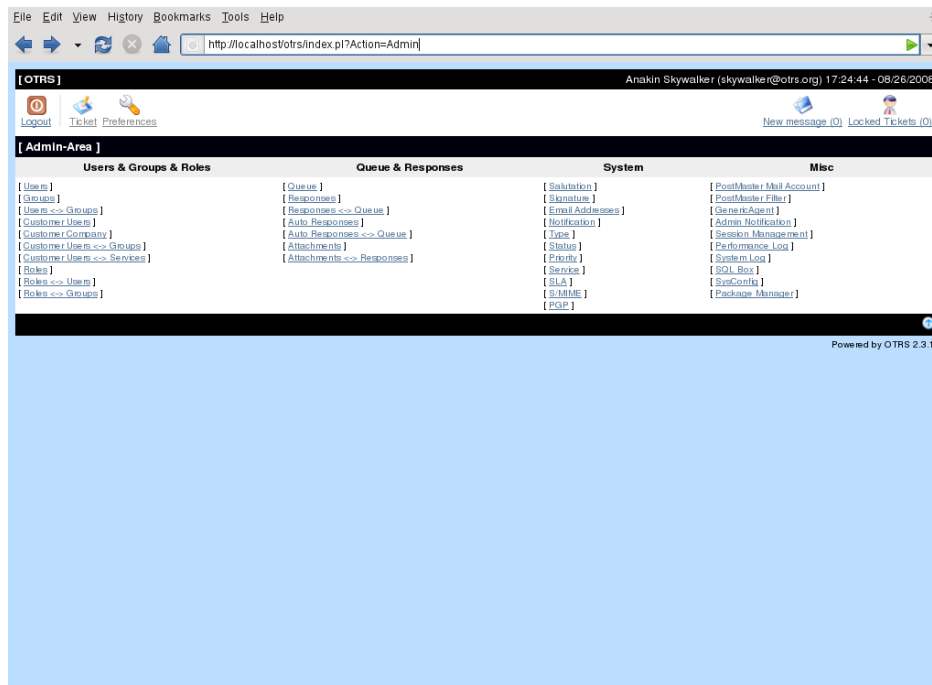
Basics

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 users, and much more.

You start the Admin area by clicking the *Admin* link in the navigation bar of the agent interface. The link is only displayed if you are a member of the *admin* group. On a new installation you can log in with the login *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 OTRS systems.



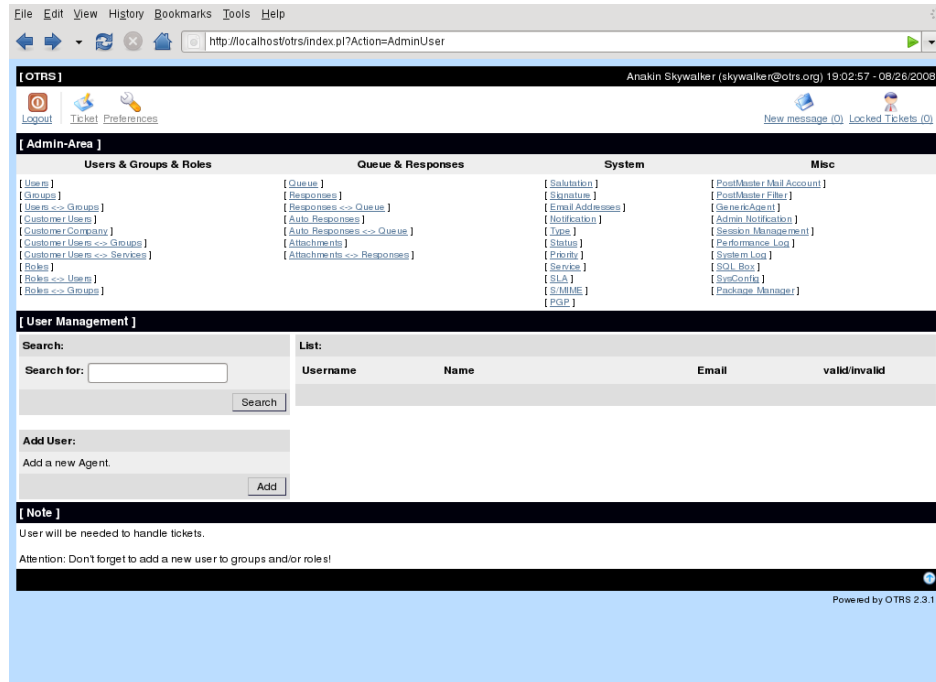
Users, groups and roles

Users

Via the link *Users* you can enter the user management of OTRS. Administrators can add, change or deactivate user accounts. Administrators can also manage user preferences, for instance the language of the user interface and notification settings for individual users.

Note

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



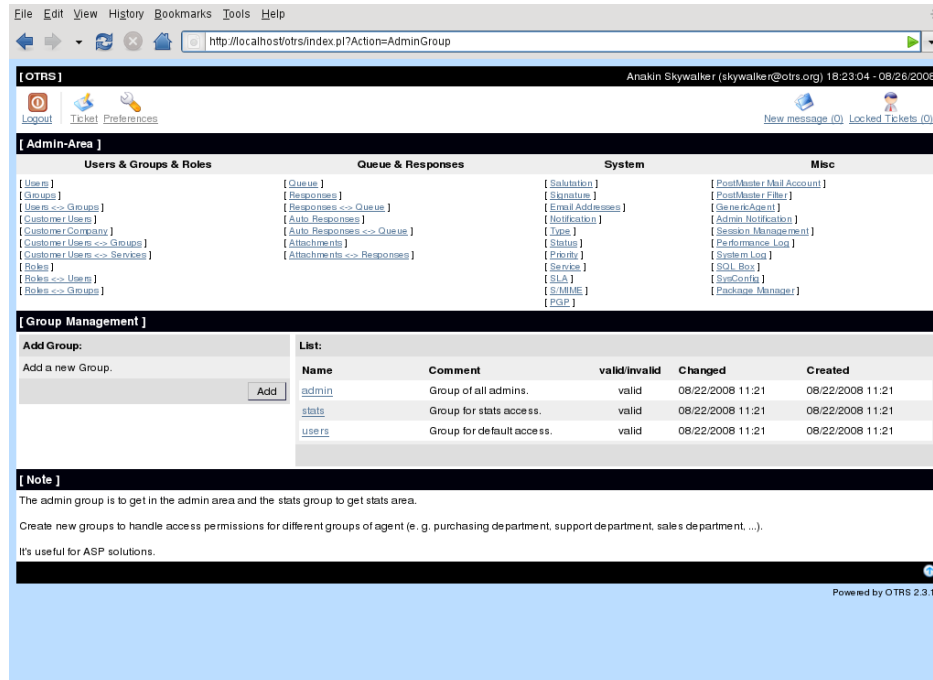
Note

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

After you have created a new user you should make it a member of one or more groups or roles.
You will be redirected automatically to the screen for the group and user management when you finished creating a user account.

Groups

Every user account should at least belong to one group or role. You can access Group management via the *groups* link.



Note

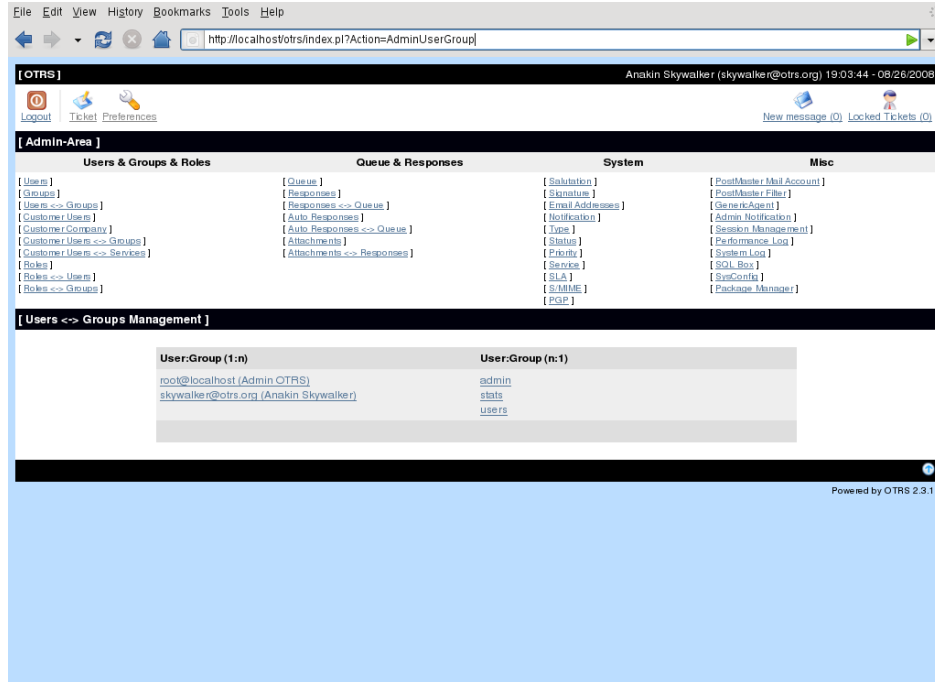
To keep the consistency of the data for OTRS it is not possible to delete a group. To deactivate a group set it to *invalid* or *invalid-temporarily*.

After installation three pre-defined groups are available.

Table 5.1. Default groups after OTRS has been installed

Group	Description
admin	Group for the users who should perform administrative tasks in the system. After the system has been installed only the user "root@localhost" is in this group.
stats	Users in this group may access the stats module of OTRS and generate statistics. After the system has been installed only "root@localhost" belongs to this group.
users	This is the group where your agents should belong to and have read and write access. If users are in this group and have write rights they can use all functions of the ticket system. After the system has been installed this group is empty.

To add a user to a group or to change the users inside a group you can use the link *Users <-> Groups*.



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

Users can have different rights in a group. Below is a list of the permission rights available:

Table 5.2. Rights in the user groups of OTRS

Right	Description
ro	Read only access to the tickets and entries or queues of this group.
move into	Right to move tickets or entries between queues or areas that belong to this group.
create	Right to create tickets or entries in the queues or areas of this group.
owner	Right to update the owner of tickets or entries in queues or areas that belong to this group.
priority	Right to change the priority of tickets or entries in queues or areas that belong to this groups.
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. The idea behind it is that the QueueView lists only the tickets the agent needs to work on. If you'd like to change this behaviour, you can set `Ticket::Frontend::AgentTicketQueue###ViewAllPossibleTickets` to `Yes`.

Roles

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

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

Note

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

The screenshot shows the OTRS Admin Area interface. At the top, there's a navigation bar with links like Logout, Ticket, and Preferences. Below that, the 'Admin-Area' is divided into four main sections: Users & Groups & Roles, Queue & Responses, System, and Misc. The 'Users & Groups & Roles' section is currently selected, showing a list of roles and groups. Below this, the 'Role Management' section is visible, featuring a form to 'Add Role' and a table to manage existing roles. The table has columns for Name, Comment, valid/invalid, Changed, and Created. A note at the bottom states: 'Create a role and put groups in it. Then add the role to the users. It's useful for a lot of users and groups.'

Note

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

The admin area of OTRS

The screenshot shows the OTRS Admin Area interface. The top navigation bar includes links for Logout, Ticket, and Preferences. The main content area is titled '[Admin-Area]' and contains a table with four columns: Users & Groups & Roles, Queue & Responses, System, and Misc. Below this table, there is a section titled '[Roles <-> Users Management]' which displays a table with two columns: 'User:Role (1:n)' and 'User:Role (n:1)'. The table lists the following roles: root@localhost (Admin OTRS) and skywalker@otrs.org (Anakin Skywalker).

Users & Groups & Roles	Queue & Responses	System	Misc
[Users]	[Queue]	[Substitution]	[PostMaster Mail Account]
[Groups]	[Responses]	[Signature]	[PostMaster Filter]
[Users <-> Groups]	[Responses <-> Queue]	[Email Addresses]	[GenericAgent]
[Customer Users]	[Auto Responses]	[Notification]	[Admin Notification]
[Customer Company]	[Auto Responses <-> Queue]	[Type]	[Session Management]
[Customer Users <-> Groups]	[Attachments]	[Status]	[Performance Log]
[Customer Users <-> Services]	[Attachments <-> Responses]	[Priority]	[System Log]
[Roles]		[Service]	[SQL Box]
[Roles <-> Users]		[SLA]	[SysConfig]
[Roles <-> Groups]		[SLIME]	[Package Manager]
		[PGP]	

[Roles <-> Users Management]

User:Role (1:n)	User:Role (n:1)
root@localhost (Admin OTRS)	
skywalker@otrs.org (Anakin Skywalker)	

Powered by OTRS 2.3.1

The screenshot shows the OTRS Admin Area interface. The top navigation bar includes links for Logout, Ticket, and Preferences. The main content area is titled '[Admin-Area]' and contains a table with four columns: Users & Groups & Roles, Queue & Responses, System, and Misc. Below this table, there is a section titled '[Roles <-> Groups Management]' which displays a table with two columns: 'Role:Group (1:n)' and 'Role:Group (n:1)'. The table lists the following groups: admin, stats, and users.

Users & Groups & Roles	Queue & Responses	System	Misc
[Users]	[Queue]	[Substitution]	[PostMaster Mail Account]
[Groups]	[Responses]	[Signature]	[PostMaster Filter]
[Users <-> Groups]	[Responses <-> Queue]	[Email Addresses]	[GenericAgent]
[Customer Users]	[Auto Responses]	[Notification]	[Admin Notification]
[Customer Company]	[Auto Responses <-> Queue]	[Type]	[Session Management]
[Customer Users <-> Groups]	[Attachments]	[Status]	[Performance Log]
[Customer Users <-> Services]	[Attachments <-> Responses]	[Priority]	[System Log]
[Roles]		[Service]	[SQL Box]
[Roles <-> Users]		[SLA]	[SysConfig]
[Roles <-> Groups]		[SLIME]	[Package Manager]
		[PGP]	

[Roles <-> Groups Management]

Role:Group (1:n)	Role:Group (n:1)
	admin
	stats
	users

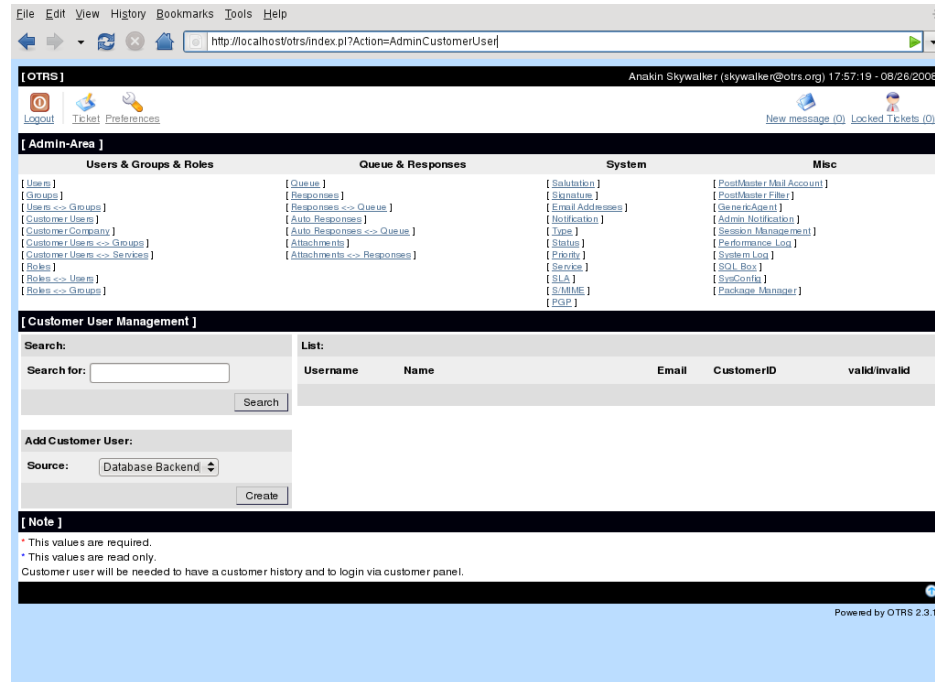
Powered by OTRS 2.3.1

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

Customer users and customer groups

Customer users

OTRS supports different types of users. Using the link "Customer Users" used to manage accounts of your customers. A customer can log in the system via the customer interface (customer.pl). The customer interface enables your customers to access their own tickets as well as log new ones. Also a customer user is needed for the ticket history in the system.



You can search for a specific customer user in the database and change the customer user back-end. More information about customer user back-ends is available in the chapter about external back-ends .

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

Note

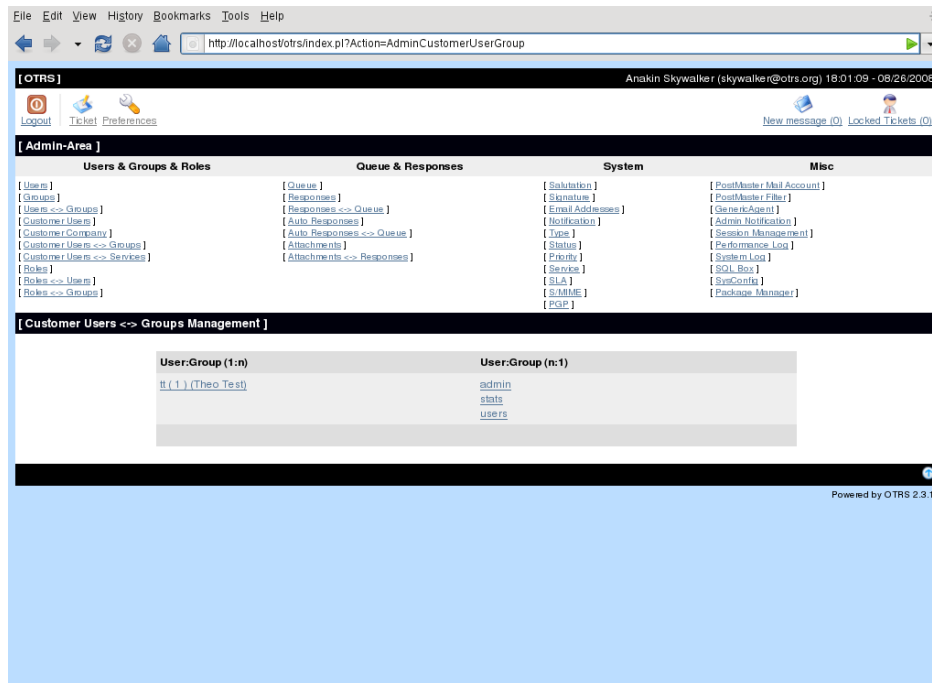
To keep the consistency of the data for OTRS it is not possible to delete a customer user. To deactivate a customer user account set it to "invalid" or "invalid-temporarily".

Customer groups

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

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

The next step is to activate the customer group support. This can be done with the configuration parameter `CustomerGroupSupport` via the graphical configuration front-end of OTRS. Using the parameter `CustomerGroupAlwaysGroups` you can specify the groups for a new added customer user, the new account will be added to this group automatically.



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

Queues

Through the link "Queue" you can manage the queues of your system. In a new installed system the "Junk", "Misc", "Postmaster" and "Raw" queues already exist. "Raw" is the default queue, all incoming messages will be stored in this queue if no filter rules are defined. The "Junk" queue can be used to store spam messages.

The screenshot shows the OTRS Admin Area interface. At the top, there's a browser window with the URL `http://localhost/otrs/index.pl?Action=AdminQueue`. The OTRS logo and user information (Anakin Skywalker) are visible. The main menu is divided into four sections: Users & Groups & Roles, Queue & Responses, System, and Misc. The Queue Management section is active, showing a 'Change' section with a list of queues (Junk, Misc, Postmaster, Raw) and a 'Add' section with various configuration fields.

Queue Management

Change:

- Junk
- Misc
- Postmaster
- Raw

Add:

Name:

Sub-Queue of:

Group:

Unlock timeout: minutes
(0 = no unlock - 24 hours = 1440 minutes - Only business hours are counted.)

Escalation - First Response Time: minutes (Notify by)
(0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.)

Escalation - Update Time: minutes (Notify by)
(0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.)

Escalation - Solution Time: minutes (Notify by)
(0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.)

Follow up Option:

Ticket lock after a follow up:

Systemaddress:

Here you can add queues and modify them. You can specify the name and the group that should use the queue. You can also specify whether you want the new queue to be a sub-queue of another already existing queue.

You can define an unlock timeout for this queue. 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.

There are three escalation time settings at the queue level. For 'Escalation - First Response Time': if there is not added a customer interaction, 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 is re-opened. This ensures that a follow up for a ticket is processed by the agent that has previously handled the same ticket.

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

Note

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

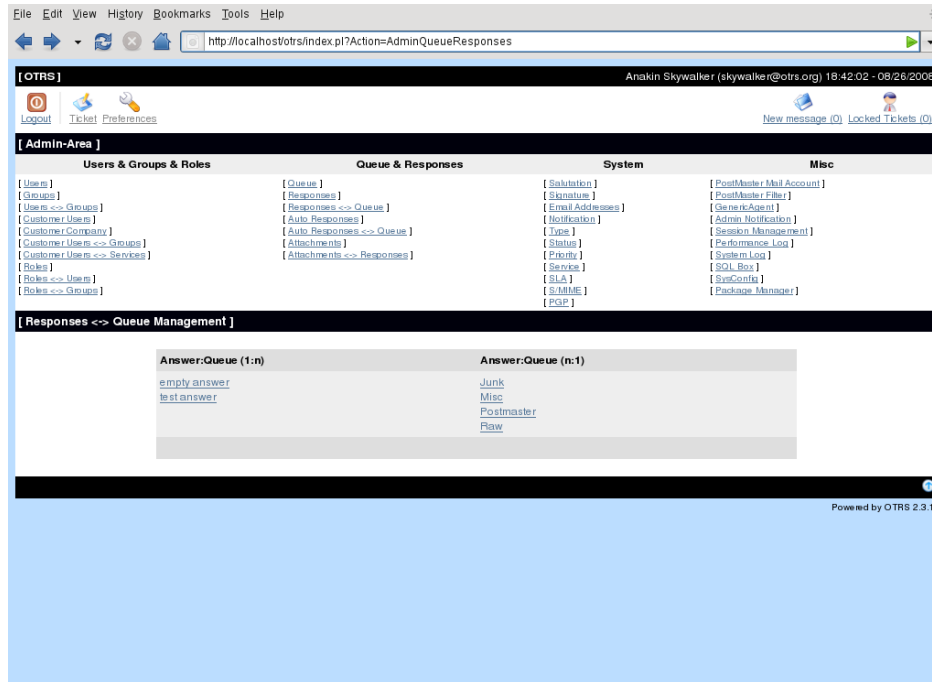
All configuration settings for new queues are also available for sub-queues.

Salutations, signatures, attachments and responses

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

The screenshot shows the OTRS AdminArea interface. At the top, there's a navigation bar with links like Logout, Ticket, and Preferences. Below this is a table with four columns: Users & Groups & Roles, Queue & Responses, System, and Misc. The Queue & Responses column is currently selected, showing a list of response-related links. Below the table is the 'Response Management' section. It has two main areas: 'Change:' on the left, which contains a list of existing responses ('empty answer (1)' and 'test answer (2)'), and 'Add:' on the right, which contains a form for creating a new response. The 'Add:' form includes fields for Name, Response (a large text area), Attachments (a dropdown menu), Valid (a dropdown menu set to 'valid'), and Comment (a text area). There are 'Change' and 'Add' buttons at the bottom of their respective sections.

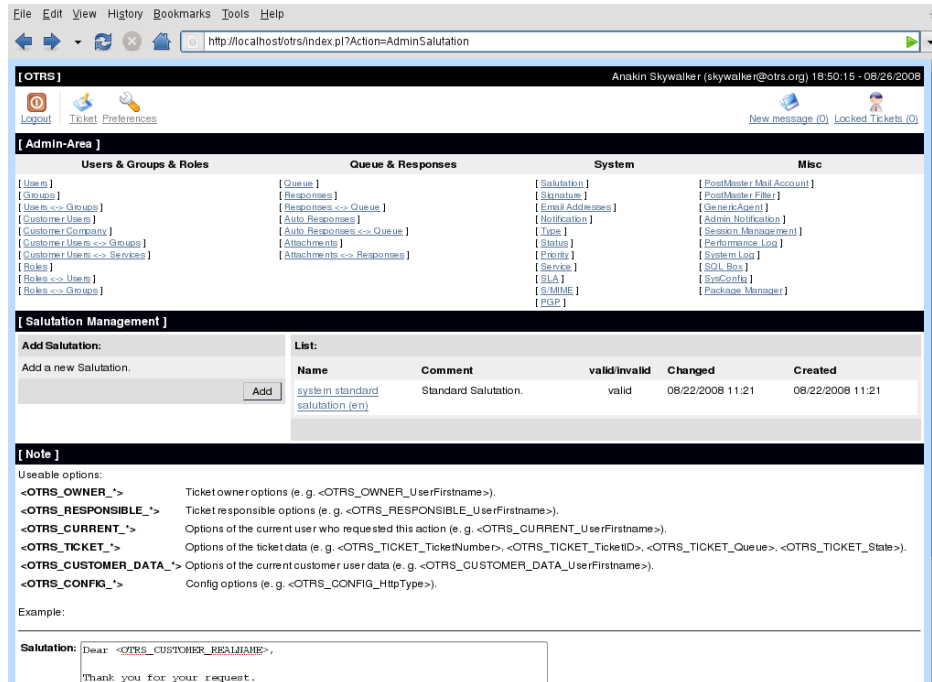


To add or remove response to one or more queues the "Responses <-> Queues" link can be used.

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

Salutations

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



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

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

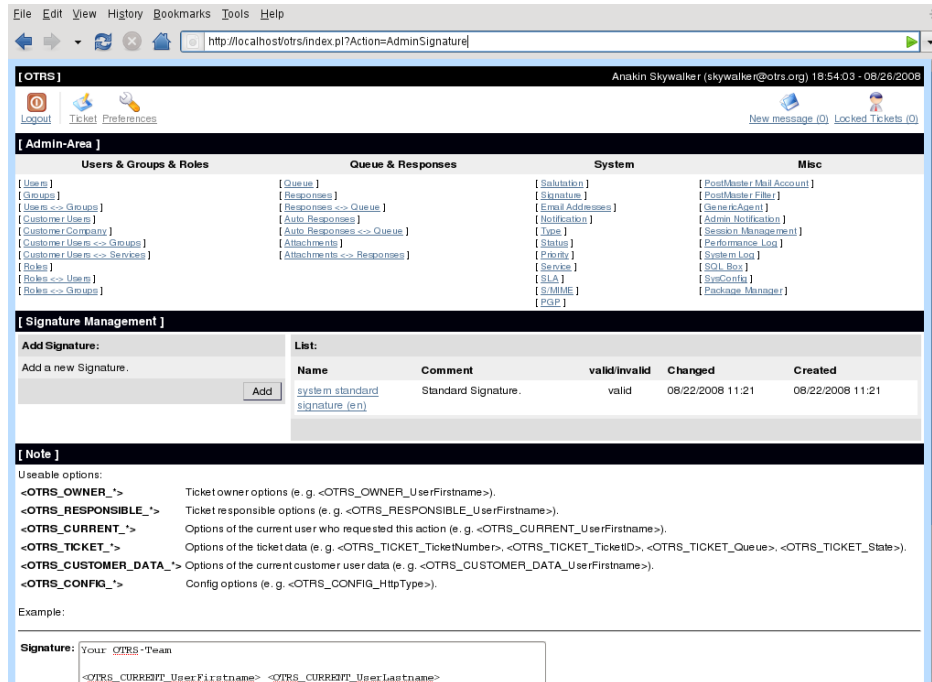
The different variables you can use in responses are listed in the lower part of the salutation screen. If you use for example the variable <OTRS_LAST_NAME> the last name of the sender of the ticket will be included in your reply.

Note

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

Signatures

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



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

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

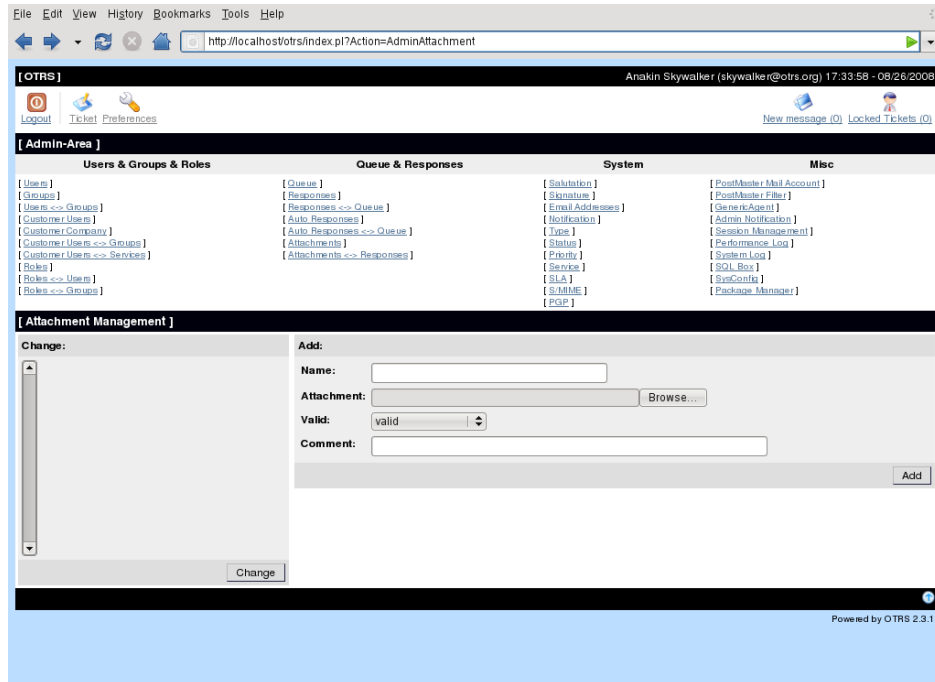
Note

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

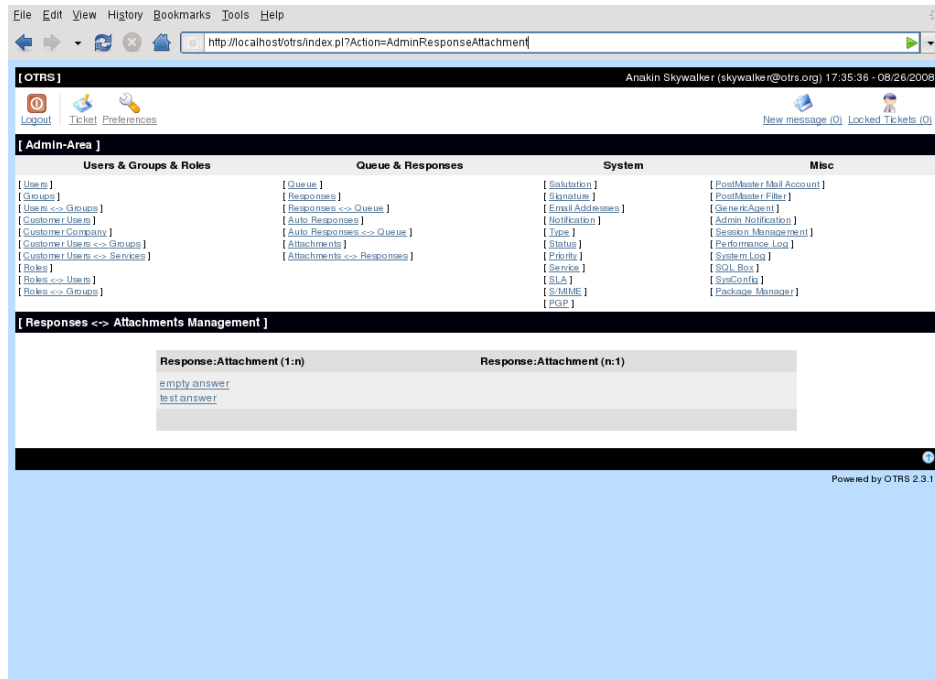
Attachments

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. The agent can remove the attachment from an individual response before sending it to the customer if this would be necessary.

The admin area of OTRS



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



Note

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

Auto answers

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

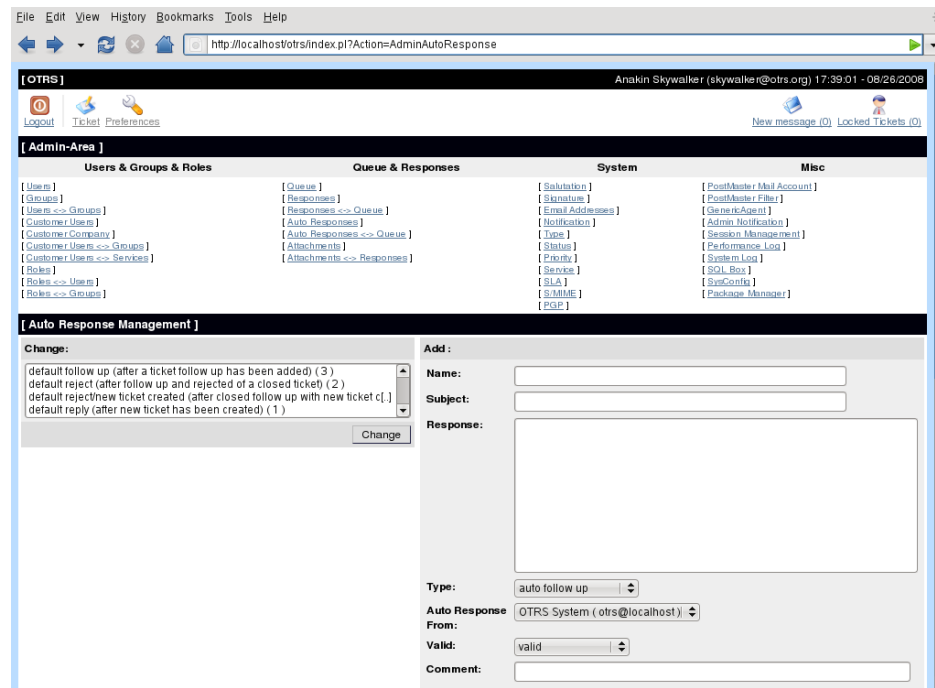


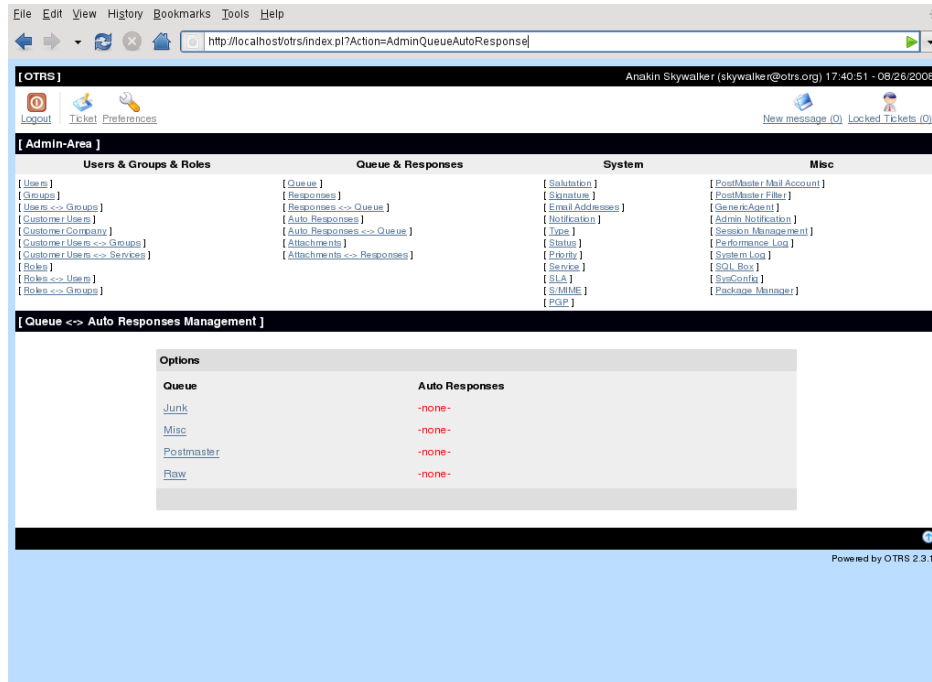
Table 5.3. Events for auto answers

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

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

Note

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



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

Email addresses

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

The admin area of OTRS

The screenshot shows the OTRS Admin Area interface. At the top, there's a navigation bar with 'Logout', 'Ticket', and 'Preferences' links. Below this, the 'Admin-Area' is divided into four main sections: 'Users & Groups & Roles', 'Queue & Responses', 'System', and 'Misc'. The 'System' section is currently active, displaying a table of system email addresses. The table has columns for 'Name', 'Realname', 'valid/invalid', 'Changed', and 'Created'. One entry is visible: 'otrs@localhost' with 'OTRS System' as the realname, 'valid' status, and creation/modification dates of '08/22/2008 11:21'. Below the table, there's a 'Note' section explaining that all email addresses get excluded on replaying on composing and email, and that incoming emails with this 'Email' (To:) will be dispatched in the selected queue. An example is provided with 'Email: support@company.com', 'Realname: Support Team', and 'Queue: Tech (1)'.

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

Note

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

Notifications

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

The screenshot shows the OTRS Admin Area interface, specifically the 'Notification Management' section. It features a list of notification events on the left, such as 'ar_SA:Agent:AddNote', 'ar_SA:Agent:Escalation', and 'bg:Agent:AddNote'. A 'Change' button is at the bottom of this list. On the right, there's a form to configure a notification, including fields for 'Type', 'Subject', and 'Text'. The 'Text' field is a rich text editor with various formatting options like bold, italic, underline, and font color.

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

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

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.

[OTRS] Anakin Skywalker (skywalker@otrs.org) 04.06.13 - 08/05/2010

Logout Dashboard Ticket State Customer Preferences Admin

New message (0) Locked Tickets (0)

[Admin-Area]

- Users & Groups & Roles**
 - [Users]
 - [Groups]
 - [Users <-> Groups]
 - [Customer Users]
 - [Customer Companies]
 - [Customer Users <-> Groups]
 - [Customer Users <-> Services]
 - [Roles]
 - [Roles <-> Users]
 - [Roles <-> Groups]
- Queue & Responses**
 - [Queue]
 - [Responses]
 - [Responses <-> Queue]
 - [Queue Responses]
 - [Auto Responses <-> Queue]
 - [Attachments]
 - [Attachments <-> Responses]
- System**
 - [Salutation]
 - [Signature]
 - [Email Addresses]
 - [Notification]
 - [Notification (Event)]
 - [Type]
 - [Status]
 - [Priority]
 - [Service]
 - [SLA]
 - [SUMTIME]
 - [ESC]
- Misc**
 - [PostMaster Mail Account]
 - [PostMaster Filter]
 - [ScenarioAgent]
 - [Admin Notification]
 - [Session Management]
 - [Performance Log]
 - [System Log]
 - [SQL Base]
 - [SysConfig]
 - [Password Manager]

[Notification Management]

Add Notification:

Add a new Notification.

List:

Name	valid/invalid	Changed	Created	Delete
My test notification	valid	11/17/2010 00:11	11/17/2010 00:11	x
My test notification 2	valid	11/17/2010 00:12	11/17/2010 00:12	x

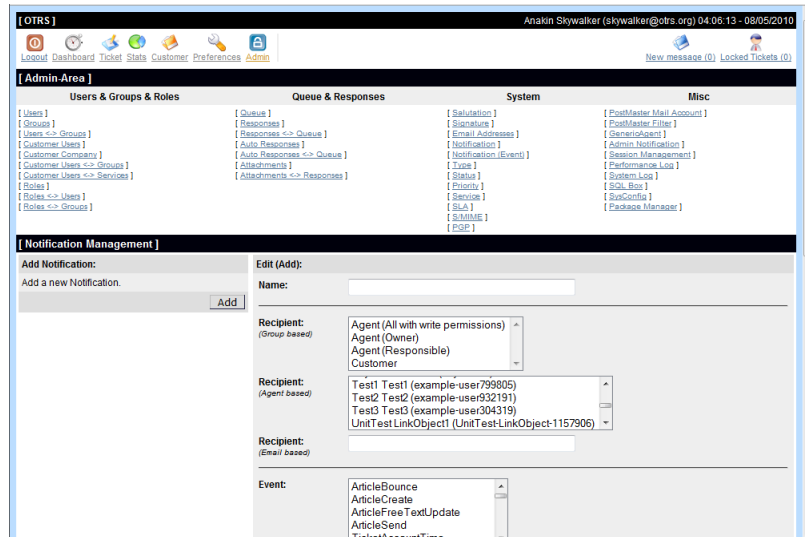
[Note]

Notifications are sent to an agent or a customer.

Useable options:

- <OTRS_AGENT_SUBJECT[20]> To get the first 20 character of the subject (of the latest agent article).
- <OTRS_AGENT_BODY[5]> To get the first 5 lines of the body (of the latest agent article).
- <OTRS_AGENT_> To get the article attribute (e. g. <OTRS_AGENT_From>, <OTRS_AGENT_To>, <OTRS_AGENT_Cc>, <OTRS_AGENT_Subject> and <OTRS_AGENT_Body>).
- <OTRS_CUSTOMER_SUBJECT[20]> To get the first 20 character of the subject (of the latest customer article).
- <OTRS_CUSTOMER_BODY[5]> To get the first 5 lines of the body (of the latest customer article).
- <OTRS_CUSTOMER_REALNAME> To get the realname of the sender (if given).
- <OTRS_CUSTOMER_> To get the article attribute (e. g. <OTRS_CUSTOMER_From>, <OTRS_CUSTOMER_To>, <OTRS_CUSTOMER_Cc>).

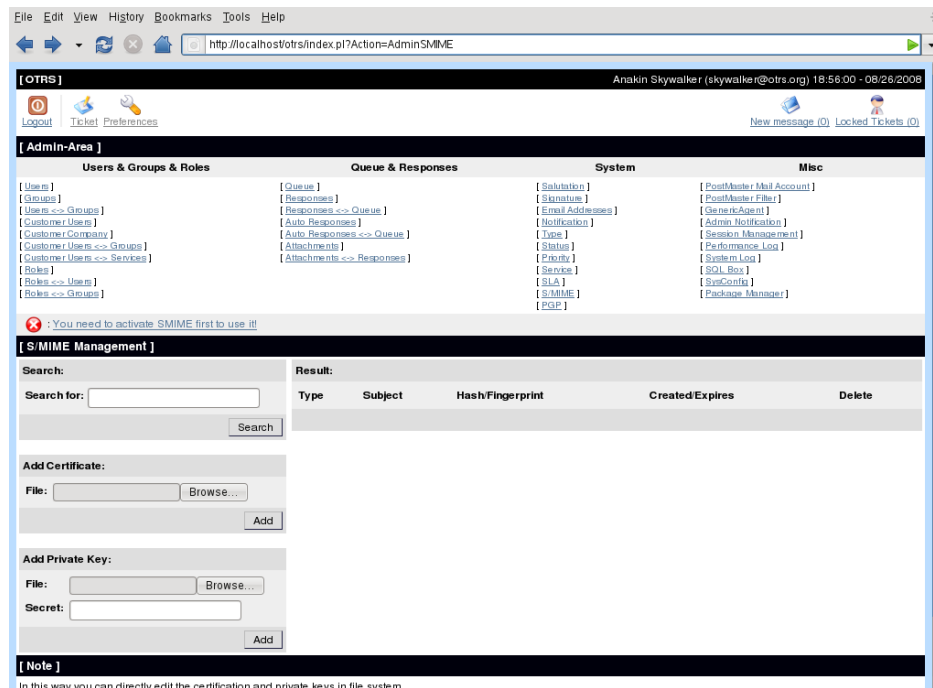
In the Event Notifications Management screen you can create a new notification by clicking on the Add button.



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.

SMIME

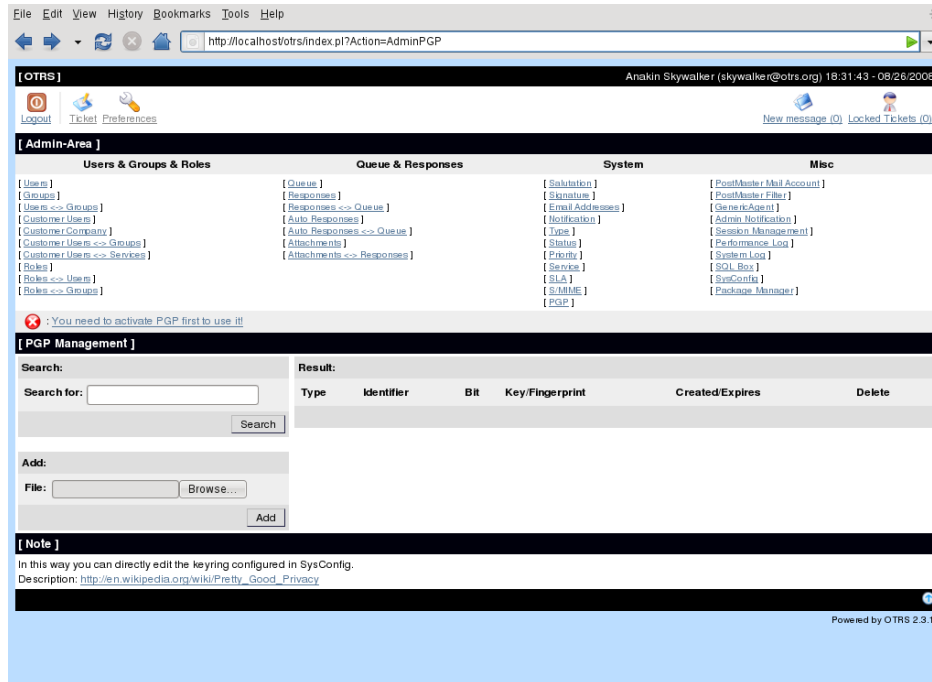
OTRS can process incoming S/MIME encoded messages. Also we can sign outgoing mails. Before this feature can be used you need to activate it and change some config parameters in the SysConfig.



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

PGP

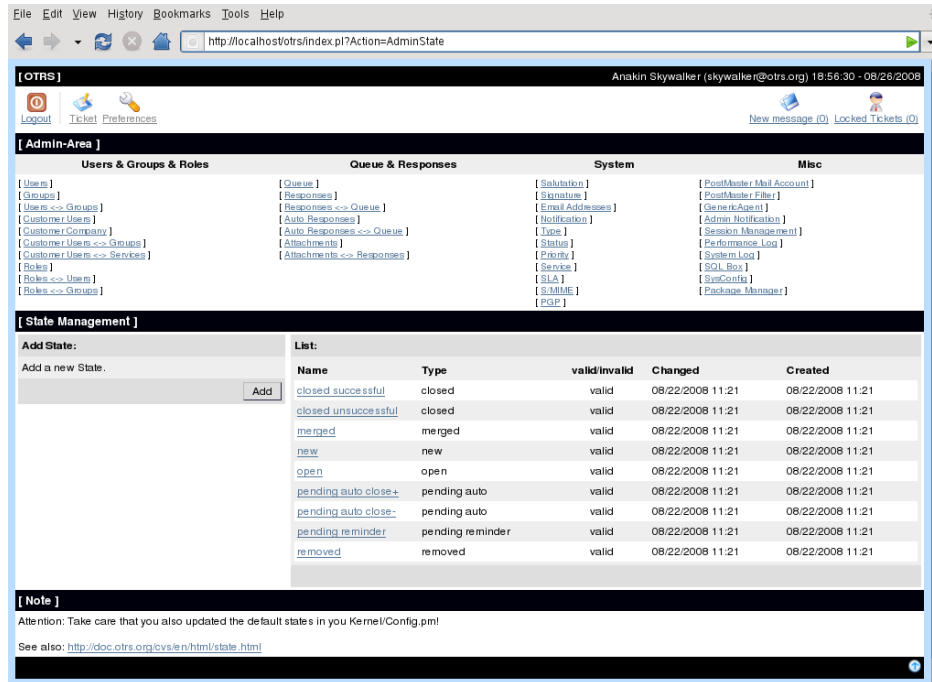
You can use OTRS to de- and encrypt messages with PGP. Also you can sign outgoing messages. Before this feature can be used you need to activate it and change some config parameters in the SysConfig.



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

States

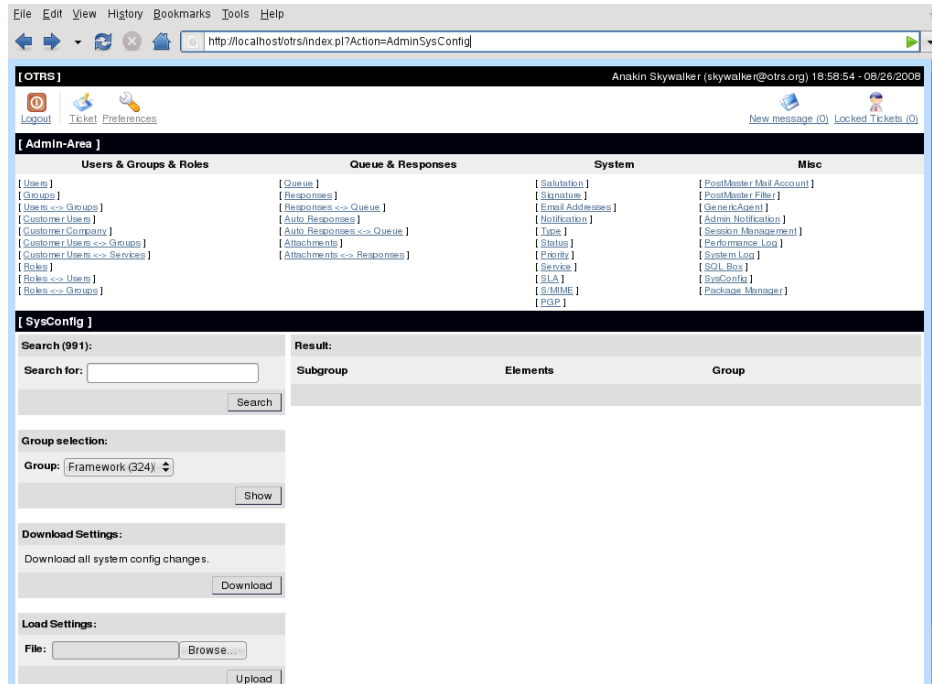
Through the "Status" link in the admin area of OTRS you can manage the different states you want to use in the ticket system.



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

SysConfig

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

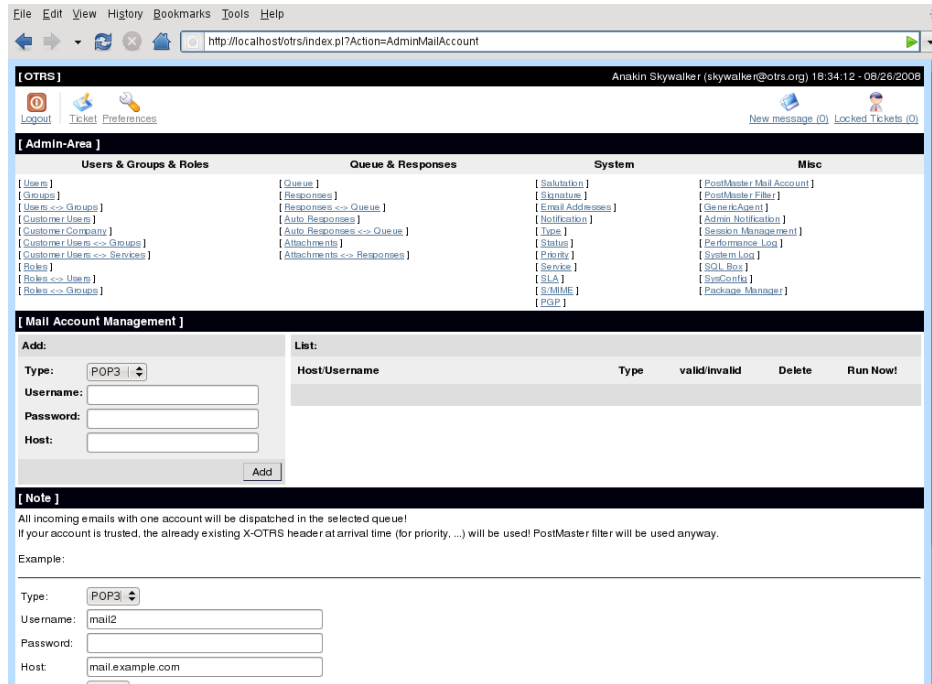


The "SysConfig" link in the admin area of OTRS loads the graphical configuration front-end. You can upload own config files for the system and backup all your current settings into a file. Almost all config parameters of the OTRS framework and the installed applications can be viewed and changed through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

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

Using mail accounts

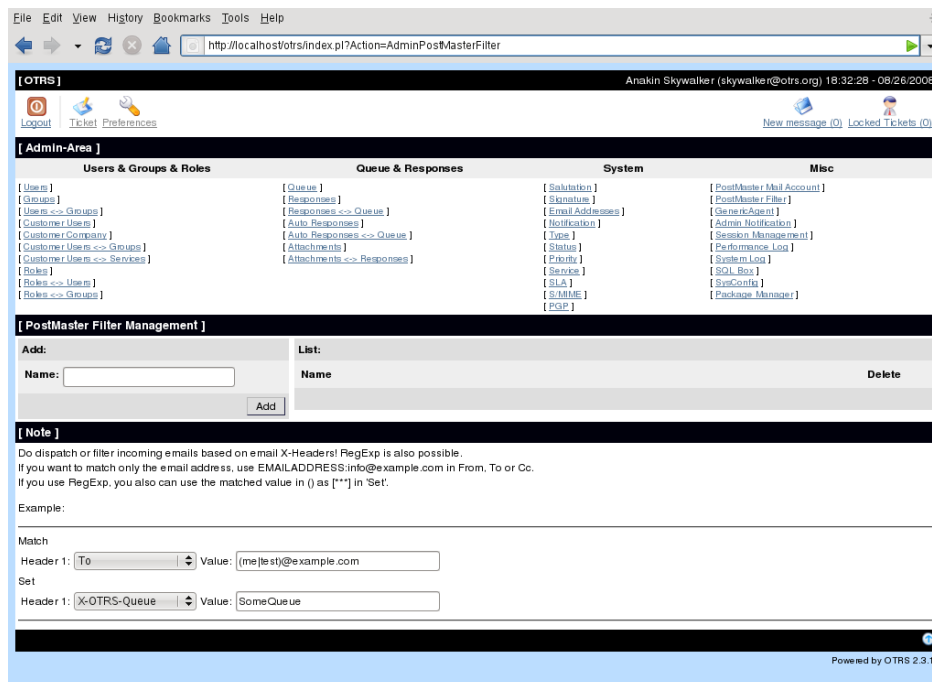
There are several possibilities to transport new emails into the ticket system. One possibility is the PostMaster.pl module that pipes the mails directly into the system. Another possibility are mail accounts which can be administrated through the web interface. The "PostMaster Mail Account" link in the admin area of OTRS loads the management console for the mail accounts. OTRS supports the mail protocols POP3, POP3S, IMAP and IMAPS.



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

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 mail accounts are used, or if `PostMaster.pl` is used to get messages into the ticket system. Filter rules can be created through the link "PostMaster Filter" in the admin area of OTRS.



A filter rule consists of one or more filter criteria that must match if the filter rule shall be executed and one or more actions that are executed, if the filter criteria match. 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 following table 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.
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. 2007-03-20 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.

Name	Possible values	Description
X-OTRS-ArticleValue(1 2 3):	Additional info value for the article.	Saves an additional info value for the article.
X-OTRS-SenderType:	agent, system, customer	Sets the type of the ticket sender.
X-OTRS-ArticleType:	email-external, email-internal, email-notification-ext, email-notification-int, phone, fax, sms, webrequest, note-internal, note-external, note-report	Sets the article type for the incoming ticket.
X-OTRS-TicketKey(1 2 ... 8):	Additional info key for the ticket.	Saves an additional info key for the ticket.
X-OTRS-TicketValue(1 2 ... 8):	Additional info value for the ticket.	Saves an additional info value for the ticket.
X-OTRS-Loop:	True	If this X-OTRS header is set no auto answer is delivered to the sender of the message (mail loop protection).

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

Example 5.1. Sort spam mails into a specific queue

A useful filter rule could be to let OTRS sort mails marked for spam with a spam detection tool such as SpamAssassin into the "Junk" queue automatically. 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 more complex systems.

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, e.g.

The admin area of OTRS

The screenshot shows the OTRS Admin Area interface. At the top, there's a navigation bar with links like Logout, Ticket, and Preferences. Below this, the 'Admin-Area' is divided into four main sections: Users & Groups & Roles, Queue & Responses, System, and Misc. The 'GenericAgent' section is active, displaying a 'Job-List' table. The table has columns for Name, Last run, valid/invalid, Delete, and Run Now!. A single job named 'Test' is listed with a 'valid' status. Below the table, there's a form to 'Add' a new job, with fields for Name, Last run, valid/invalid, Delete, and Run Now!.

To create a new job for the GenericAgent the link "GenericAgent" in the admin area of OTRS can be used. A table with the already created jobs is displayed where jobs can be executed manually or be removed. To create a new job a name must be specified and the "Add" button must be pressed.

This screenshot is similar to the previous one, but it highlights the 'Add' button in the 'Job-List' table. The 'Add' button is located at the bottom right of the table, next to the 'Run Now!' column. The 'Test' job is still listed in the table.

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

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

Admin email

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

The screenshot shows the OTRS Admin Area interface. At the top, there's a navigation bar with links like Logout, Ticket, and Preferences. Below it, there's a table with four columns: Users & Groups & Roles, Queue & Responses, System, and Misc. The 'System' column contains links for various system settings. Below the table is a form to add a new system address, with fields for Name, Realname, valid/invalid, Changed, and Created. At the bottom, there is a 'Note' section and a form to specify the sender, subject, and message text of the notification.

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

Session management

You can see all logged in users and their session details by clicking the *Session Management* link in the admin area.

The screenshot shows the OTRS Admin Area interface. At the top, there's a navigation bar with links like Logout, Ticket, and Preferences. Below this, the 'Admin-Area' is divided into four main sections: Users & Groups & Roles, Queue & Responses, System, and Misc. The 'Session Management' section is active, displaying an overview of sessions and a detailed list of active sessions.

Session Management Overview:

- All Sessions: 2
- Agent Sessions: 2
- Customer Sessions: 0
- Uniq Agent: 1
- Uniq Customer: 0

Session List:

Session	Content	kill session
31506a1b75147671eed5b29e371a404a263	LastNavBarName=Admin; LastScreenOverview=Action=AgentTicketMailbox; LastScreenView=Action=AgentTicketMailbox; UseComments; UseCreateNextMask; UseEmail=skywalker@ots.org; UseFirstname=Anakin; UseId=3; UseRoleIdGroupRole[admin]=Yes; UseRoleIdGroupRole[user]=Yes; UseRoleIdGroupRole[admin]=Yes; UseRoleIdGroupRole[user]=Yes; UseLanguage=en; UseLastRequest=1219771845; UseLastname=Skywalker; UseLogin=skywalker@ots.org; UsePw=xxxxxxxx; UseQueueView=AgentTicketQueueTicketView; UseQueueViewShowTickets=15; UseRefreshTime=2; UseRemoteAddr=88.78.21.146; UseRemoteUseAgent=Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US; rv:1.9.0.1) Gecko/2008070208 Firefox/3.0.1; UseSalutation=; UseSendFollowUpNotification=0; UseSendLockTimeOutNotification=0; UseSendNewNotification=0; UseSendNewTicketNotification=0; UseSessionStart=2008-08-26 12:40:48 / 6 h; UseSpeedDial=deutch; UseTheme=Standard; UserType=User; ValidId=1;	X
3152728ddc04cd8e7a82cc34fc6bda14158	LastNavBarName=Preferences; LastScreenOverview=8; LastScreenView=8; UseComments; UseCreateNextMask; UseEmail=skywalker@ots.org; UseFirstname=Anakin; UseId=3; UseRoleIdGroupRole[admin]=Yes; UseRoleIdGroupRole[user]=Yes; UseRoleIdGroupRole[admin]=Yes; UseRoleIdGroupRole[user]=Yes; UseLanguage=en; UseLastLogin=1219754642; UseLastPw=xxxxxxxx; UseLastRequest=1219776808; UseLastname=Skywalker; UseLogin=skywalker@ots.org; UsePw=xxxxxxxx; UseQueueView=AgentTicketQueueTicketView; UseQueueViewShowTickets=15; UseRefreshTime=2; UseRemoteAddr=88.78.21.146;	X

Some statistics about all active sessions are displayed, e.g. how many agents and customer users are logged in, and how many sessions there are. You have the option to *Kill all sessions* which can be useful if you would like to bring the system down. Also detailed information for every session is available. Every individual session can be removed separately by clicking on the x in the right-hand side of the session list.

System Log

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

The screenshot shows the OTRS Admin Area web interface. The browser address bar displays the URL `http://localhost/otrs/index.pl?Action=AdminLog`. The page title is `[Admin-Area]`. Below the title bar, there are navigation links: `Logout`, `Ticket`, `Preferences`, `New message (0)`, and `Locked Tickets (0)`. The main content area is divided into four sections: `Users & Groups & Roles`, `Queue & Responses`, `System`, and `Misc`. The `System` section is currently selected, displaying a list of system components. Below this, there is a `System Log` table with columns: `Time`, `Priority`, `Facility`, and `Message`. The log contains several entries, including session removals, user authentication, and ticket creation. One entry is highlighted in red, indicating an error: `Email address (tt@example.com) not valid (invalid tt@example.com (config))!`.

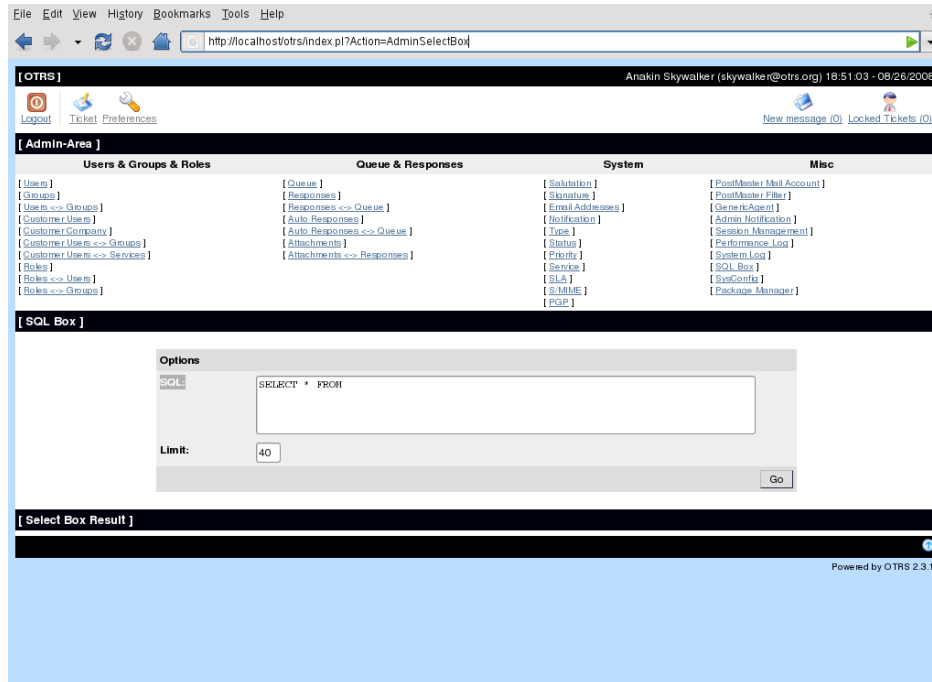
Each line in the log contains a time stamp, the log priority, the system component and the log entry itself.

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.

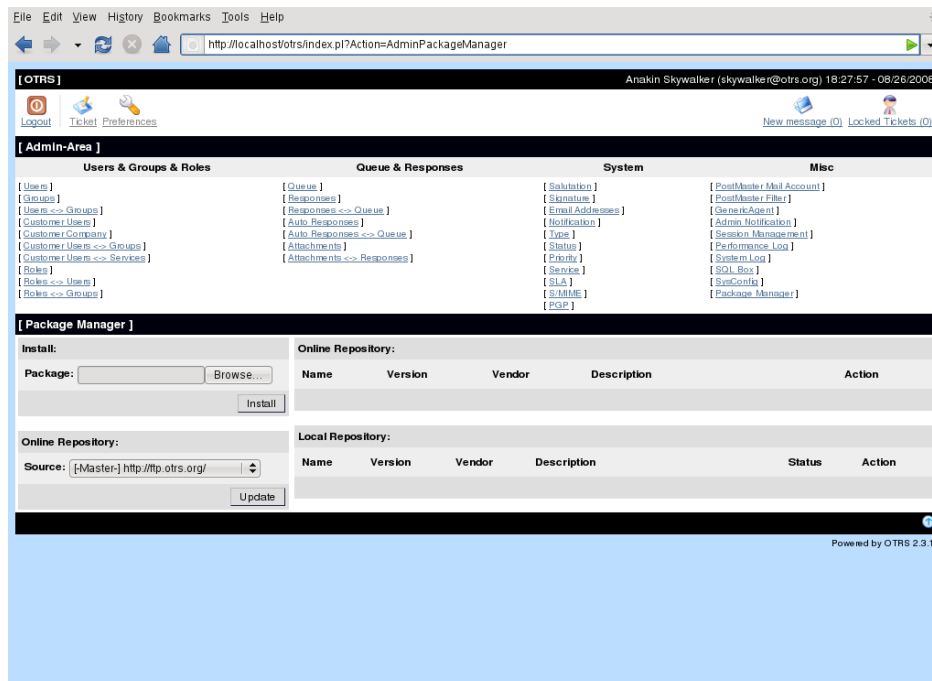
SQL queries via the SQL box

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



Package manager

With the Package Manager you can install and manage packages that extend the functionality of OTRS. See the Additional applications section for a discussion on the extensions that are available from the OTRS repositories.



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* button. The right side of the screen shows the available packages. You can install a package by clicking on the *Install* button. 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 installed locally. Just click Upgrade and it will install the 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 next to the *Package* field and select the .opm file on your disk. Click *Open* and then *Install*. 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, for instance 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. Configuring the system

The config files of OTRS

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

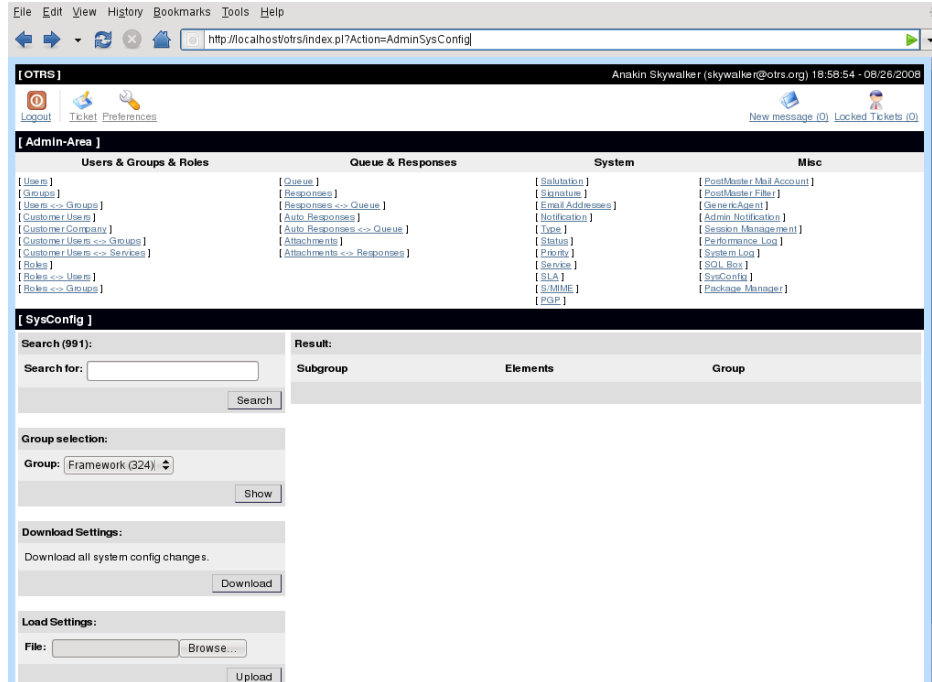
The file `Kernel/Config/Defaults.pm` contains the parameters of the central OTRS framework. All basic system settings like the mail configuration, database connection, default charset or standard language are located in this file. The file `Kernel/Config/Files/Ticket.pm` contains all configuration parameters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

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

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

Configuring the system through the web interface

With OTRS 2.0 and subsequent versions nearly all configuration parameters of the central framework or additionally installed applications can be changed easily with the graphical configuration frontend. Log in as OTRS administrator and follow the "SysConfig" link in the adminarea to execute the new configuration tool.



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

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

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

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

Note

The configuration parameters for the database connection cannot be changed via the graphical configuration frontend for security reasons. They have to be set manually in `Kernel/Config.pm`.

Chapter 7. Sending/Receiving emails

Sending emails

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)

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.

Receiving emails

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 admin interface (PostMaster Mail Account).

The screenshot shows the OTRS Admin Mail Account Management interface. The browser address bar displays `http://localhost/otrs/index.pl?Action=AdminMailAccount`. The page title is "OTRS" and the user is "Anakin Skywalker (skywalker@otrs.org) 18:34:12 - 08/26/2008". The interface includes a navigation bar with "Logout", "Ticket", and "Preferences" links. The main content area is divided into four sections: "Users and Groups & Roles", "Queue & Responses", "System", and "Misc". The "Mail Account Management" section is active, showing a form to add a new mail account. The form includes fields for "Type" (set to POP3), "Username", "Password", and "Host". Below the form is a table with columns "Host/Username", "Type", "valid/invalid", "Delete", and "Run Now!". A "Note" section at the bottom provides instructions on how incoming emails are dispatched and how the PostMaster filter is used.

[Admin-Area]

Users and Groups & Roles

- [Users]
- [Groups]
- [Users <=> Groups]
- [Customer Users]
- [Customer Company]
- [Customer Users <=> Groups]
- [Customer Users <=> Services]
- [Roles]
- [Roles <=> Users]
- [Roles <=> Groups]

Queue & Responses

- [Queue]
- [Responses]
- [Responses <=> Queue]
- [Auto Responses]
- [Auto Responses <=> Queue]
- [Attachments]
- [Attachments <=> Responses]

System

- [Solution]
- [Signature]
- [Email Addresses]
- [Notification]
- [Type]
- [Status]
- [Priority]
- [Service]
- [SLA]
- [S/MIME]
- [POP]

Misc

- [PostMaster Mail Account]
- [PostMaster Filter]
- [Generate Agent]
- [Admin Notification]
- [Session Management]
- [Performance Log]
- [System Log]
- [SQL Box]
- [SysConfig]
- [Package Manager]

[Mail Account Management]

Add:

Type:

Username:

Password:

Host:

List:

Host/Username	Type	valid/invalid	Delete	Run Now!
---------------	------	---------------	--------	----------

[Note]

All incoming emails with one account will be dispatched in the selected queue!
If your account is trusted, the already existing X-OTRS header at arrival time (for priority, ...) will be used! PostMaster filter will be used anyway.

Example:

Type:

Username:

Password:

Host:

If a new mail account is created 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.

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 this table 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 `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/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. See the `cron jobs` chapter for more information on modifying cron jobs.

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

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

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

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

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

Procmail is a very common e-mail filter in 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`. Add the following:

```

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

```

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

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

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

Example 7.1. .fetchmailrc

```

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

```

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

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

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

If you use the `bin/PostMaster.pl` or `bin/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 following example jobs can be inserted in Kernel/Config.pm

Example 7.2. Example jobs for the filter module Kernel::System::PostMaster::Filter::Match

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

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

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

Example 7.3. Example job for the filter module Kernel::System::PostMaster::Filter::CMD

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

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

Chapter 8. Time related functions

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 default business hours and holidays for your system via the SysConfig interface. in Framework > Core::Time and additionally you can define different sets of business hours, holidays and time zones as separate 'Calendars' in Framework > Core::Time::Calendar1 until Framework > Core::Time::Calendar9. You can define calendars on a queue level and on the SLA level. This means that 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. Also, you can 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.

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 send and tickets will be unlocked.

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 the corresponding section for the calendars 1-9.

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

Note

By default, OTRS ships with the *German* holidays installed. You would need to replace these with the days agreed on in your SLA, if you have one, or to the days that are considered holidays in your company.

TimeVacationDaysOneTime

Holidays that do not fall on a fixed date each year, 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.

You should use the days agreed on in your SLA, if you have one, or to the days that are considered holidays in your company.

Automated Unlocking

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

The amount of time before a ticket is unlocked can be specified in the `settings` for every queue. The module `bin/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 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, of 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 a new article to the ticket, either via the web or via an 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.

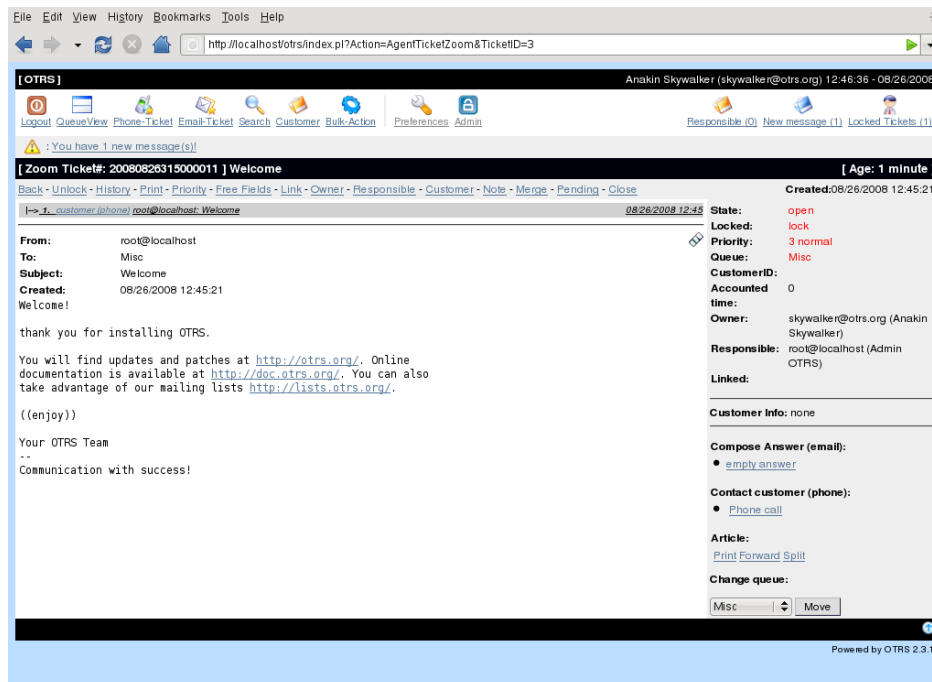
Chapter 9. Ticket responsibility and ticket watching

From OTRS 2.1 on it is possible to define a person responsible for a ticket additionally to the ticket 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.

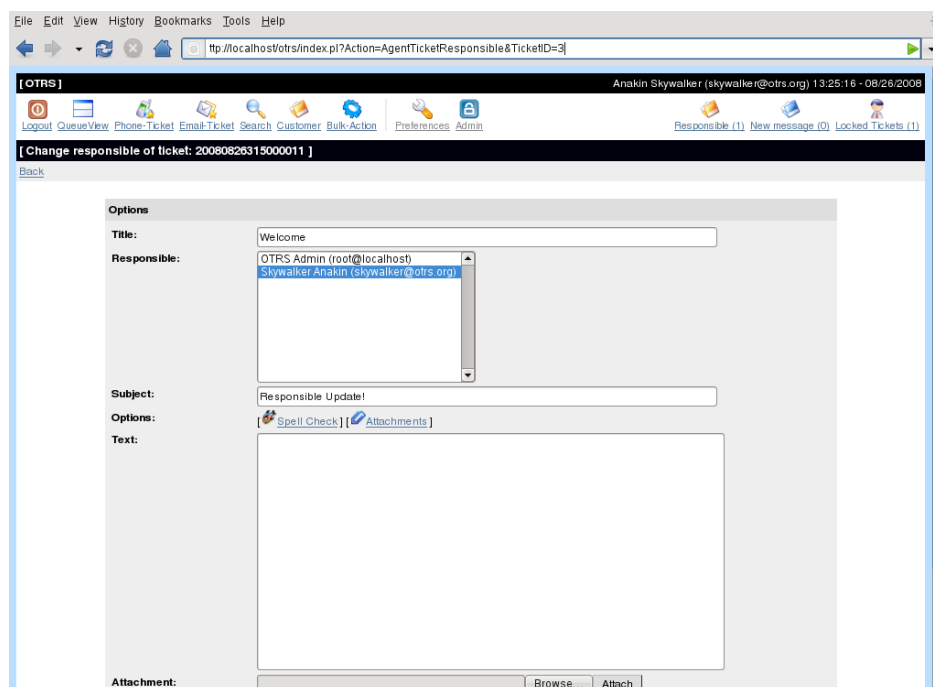
Ticket responsibility

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.



Ticket responsibility can be assigned by calling up the ticket content and activating the "responsible" switch in the ticket activities menu.



The ticket responsible template can also be used to send a message to the new agent responsible.

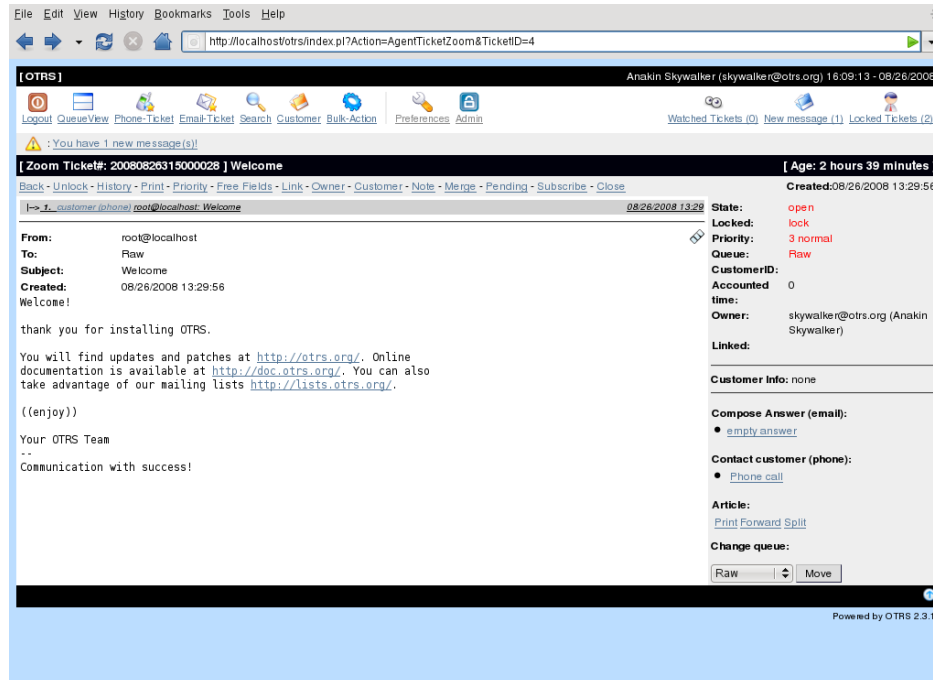
The list of all tickets for which an agent is responsible can be accessed via the "responsible" notification, which is shown in the upper part of the OTRS user interface as soon as the ticket responsible feature is activated.

Ticket watching

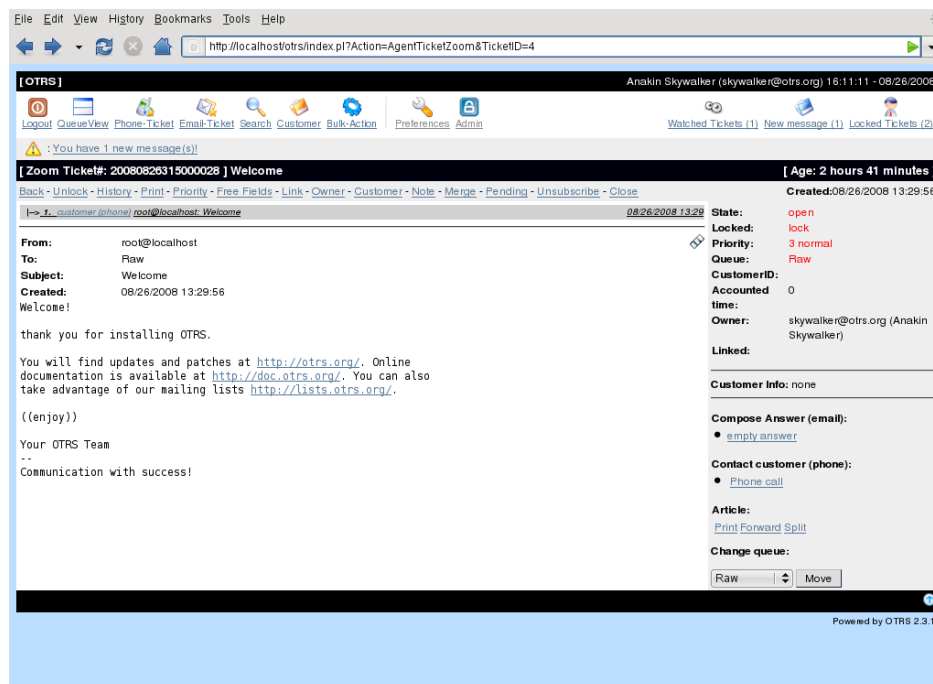
From OTRS 2.1 on, users as for example heads of department can watch certain tickets within the system without processing them by using the TicketWatcher feature.

The TicketWatcher feature can be activated with the configuration parameter `Ticket::Watcher`. Using `Ticket::WatcherGroup` one or more user groups with permission to watch tickets can be defined.

Ticket responsibility and ticket watching

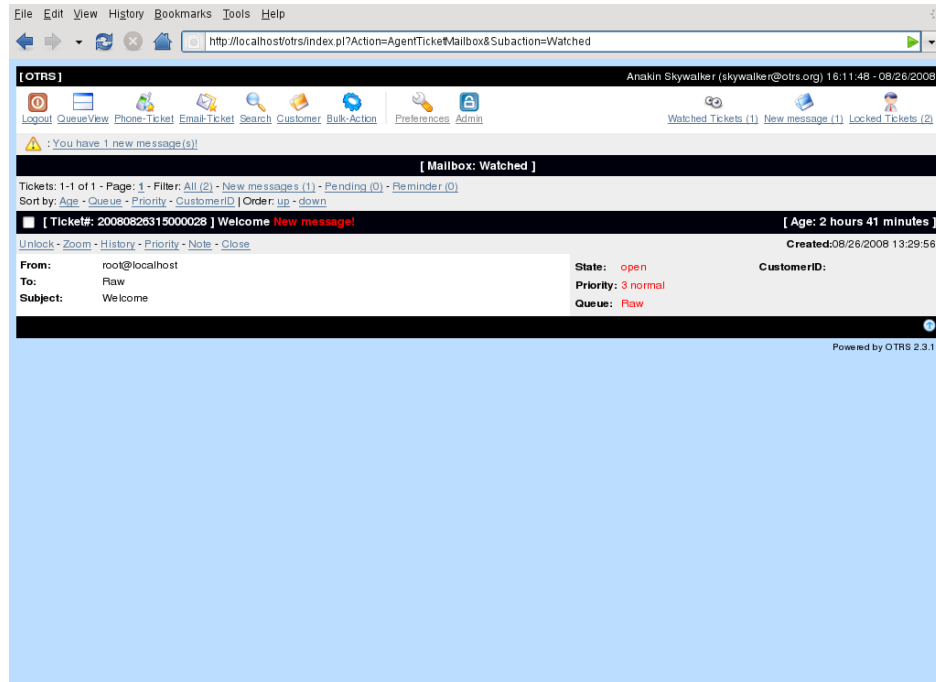


In order to watch a ticket, select the ticket content display and activate the "subscribe" switch in the ticket activities menu.



Stop watching a ticket by selecting the ticket content display and activating the "unsubscribe" switch in the ticket activities menu.

Ticket responsibility and ticket watching



A list of all watched tickets can be called up via the "Watched Tickets" notification , which is displayed in the upper part of the OTRS user interface as soon as the TicketWatcher feature is activated.

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` defines the maximum number of pages for a pdf file, which is useful if a user would 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 package and can easily be installed using the respective package manager. In case this is not possible they have to be installed with CPAN. The chapter "Installation of Perl modules" describes the installation of the Perl modules.

Chapter 11. Using external backends

Customer data

OTRS is able to handle different customer data attributes, e.g. login, email, phone number, room. This information will be displayed in both the agents and the customer frontend. Also these attributes are used for the authentication of customer users.

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

Necessary data needed for customer authentication

- User login
- Email address
- Customer ID

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

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

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 this type of backend.

Database (Default)

Example 11.1. Configuring a DB customer backend

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

```
# CustomerUser
# (customer user database backend and settings)
$Self->{'CustomerUser'} = {
    Name => '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-link, readonly, http-link-target
    [ 'UserSalutation', 'Salutation', 'salutation', 1, 0,
'var', '', 0 ],
    [ 'UserFirstname', 'Firstname', 'first_name', 1, 1,
'var', '', 0 ],
    [ 'UserLastname', 'Lastname', 'last_name', 1, 1,
'var', '', 0 ],
    [ 'UserLogin', 'Username', 'login', 1, 1,
'var', '', 0 ],
    [ 'UserPassword', 'Password', 'pw', 0, 0,
'var', '', 0 ],
    [ 'UserEmail', 'Email', 'email', 1, 1,
'var', '', 0 ],
    [ 'UserEmail', 'Email', 'email',
1, 1, 'var', '$Env{"CGIHandle"}?
```

```
Action=AgentTicketCompose&ResponseID=1&TicketID=
$Data{"TicketID"}&ArticleID=$Data{"ArticleID"}', 0 ],
    [ '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 => {
        UserSalutation => {
            'Mr.' => 'Mr.',
            'Mrs.' => 'Mrs.',
        },
    },
};
```

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

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

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

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

Database changed
mysql> ALTER TABLE customer_user ADD room VARCHAR (250);
Query OK, 1 rows affected (0.01 sec)
```

```
Records: 1 Duplicates: 0 Warnings: 0
```

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

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

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

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

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 user, the company ticket feature of OTRS is used. Company tickets can be accessed via the "Company Ticket" link in the customer panel.

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

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

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

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

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

mysql> quit
Bye
linux:~#
```

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

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

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

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

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

Example 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;".

LDAP

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

Example 11.3. Configuring a LDAP customer backend

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

```
# CustomerUser
# (customer user ldap backend and settings)
$Self->{CustomerUser} = {
    Name => 'LDAP 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=user)'
```

```

        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
perl doc 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, readonly
        [ 'UserSalutation', 'Title', 'title', 1, 0,
'var', '', 0 ],
        [ 'UserFirstname', 'Firstname', 'givenname', 1, 1,
'var', '', 0 ],
        [ 'UserLastname', 'Lastname', 'sn', 1, 1,
'var', '', 0 ],
        [ 'UserLogin', 'Username', 'uid', 1, 1,
'var', '', 0 ],
        [ 'UserEmail', 'Email', 'mail', 1, 1,
'var', '', 0 ],
        [ 'UserCustomerID', 'CustomerID', 'mail', 0, 1,
'var', '', 0 ],
    ],

```



```
#          [ 'UserCustomerIDs', 'CustomerIDs',  
'second_customer_ids', 1, 0, 'var', '', 0 ],  
          [ 'UserPhone',      'Phone',      'telephonenumber', 1, 0,  
'var', '', 0 ],  
          [ 'UserAddress',    'Address',    'postaladdress', 1, 0,  
'var', '', 0 ],  
          [ 'UserComment',    'Comment',    'description', 1, 0,  
'var', '', 0 ],  
          ],  
    };
```

If additional customer 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.

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

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 user, the company ticket feature of OTRS is used. Company tickets can be accessed via the "Company Ticket" link in the customer panel.

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

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

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

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

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.

Use more than one customer backend with OTRS

If you want to use 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".

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 user 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-link, readonly, http-link-target
    [ 'UserSalutation', 'Salutation', 'salutation', 1, 0,
'var', '', 0 ],
    [ 'UserFirstname', 'Firstname', 'first_name', 1, 1,
'var', '', 0 ],
    [ 'UserLastname', 'Lastname', 'last_name', 1, 1,
'var', '', 0 ],
    [ 'UserLogin', 'Username', 'login', 1, 1,
'var', '', 0 ],
    [ 'UserPassword', 'Password', 'pw', 0, 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 => {
    UserSalutation => {
        'Mr.' => 'Mr.',
        'Mrs.' => 'Mrs.',
    },
},
};

# 2. Customer user backend: LDAP
# (customer user 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=user)'
        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-link, readonly
#          [ 'UserSalutation', 'Title', 'title', 1, 0,
#          'var', '', 0 ],
#          [ 'UserFirstname', 'Firstname', 'givenname', 1, 1,
#          'var', '', 0 ],
#          [ 'UserLastname', 'Lastname', 'sn', 1, 1,
#          'var', '', 0 ],
#          [ 'UserLogin', 'Username', 'uid', 1, 1,
#          'var', '', 0 ],
#          [ 'UserEmail', 'Email', 'mail', 1, 1,
#          'var', '', 0 ],
#          [ 'UserCustomerID', 'CustomerID', 'mail', 0, 1,
#          'var', '', 0 ],
#          [ 'UserCustomerIDs', 'CustomerIDs',
#          'second_customer_ids', 1, 0, 'var', '', 0 ],
#          [ 'UserPhone', 'Phone', 'telephonenumber', 1, 0,
#          'var', '', 0 ],
#          [ 'UserAddress', 'Address', 'postaladdress', 1, 0,
#          'var', '', 0 ],
#          [ 'UserComment', 'Comment', 'description', 1, 0,
#          'var', '', 0 ],
#          ],
#      };
```

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

Backends to authenticate agents and customer users

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

Authentication backends for agents

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

Example 11.6. Authenticate agents against a DB backend

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

LDAP

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

Example 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 following configuration parameters 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',
];
```

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.

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';
```

Radius

The following configuration parameters 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';
```

Authentication backends for customer users

Database (Default)

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

Example 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=customerdbhost";  
# $Self->{'Customer::AuthModule::DB::User'} = "some_user";  
# $Self->{'Customer::AuthModule::DB::Password'} = "some_password";
```

LDAP

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

Example 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,
};
```

HTTPBasicAuth for customer users

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

Example 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-
authorised-for-otrs.html';
$Self->{CustomerPanelLogoutURL} = 'http://host.example.com/thanks-
for-using-otrs.html';
```

Radius

The following settings can be used to authenticate your customer users 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';
```

Customize the customer self registration

It is possible to customize the self registration for new customer users which is reachable via the customer.pl panel. New optional or required fields like 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 user.

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 128.

```
[...]
<tr>
  <td>$Text{"Room Number"}: </td>
  <td><input type="text" name="Room" value="$QData{"UserRoom"}"
size="20" maxlength="50"></td>
</tr>
[...]
```

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:

```
# CustomerUser
# (customer user database backend and settings)
$Self->{CustomerUser} = {
    Name => 'Database Backend',
    Module => 'Kernel::System::CustomerUser::DB',
    Params => {
        # if you want to use an external database, add the
        # required settings
        DSN => 'DBI:odbc:yourdsn',
        DSN =>
        'DBI:mysql:database=customerdb;host=customerdbhost',
        User => '',
        Password => '',
        Table => 'customer_user',
    },
    # customer 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-link, readonly, http-link-target
    [ 'UserSalutation', 'Salutation', 'salutation', 1, 0,
'var', '', 0 ],
    [ 'UserFirstname', 'Firstname', 'first_name', 1, 1,
'var', '', 0 ],
    [ 'UserLastname', 'Lastname', 'last_name', 1, 1,
'var', '', 0 ],
    [ 'UserLogin', 'Username', 'login', 1, 1,
'var', '', 0 ],
    [ 'UserPassword', 'Password', 'pw', 0, 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 => {
        UserSalutation => {
            'Mr.' => 'Mr.',
            'Mrs.' => 'Mrs.',
        },
    },
};
```

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. 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:~#
```

Now you took all the steps involved, and the new field for the room should be displayed in the customer.pl panel. New customer users 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

Predefined states

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

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

New

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

Open

The default state for tickets assigned to queues and agents.

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.

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.

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.

Merged

State for tickets that have been merged with other tickets.

Closed Successful

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

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.

Customizing states

Every state has a name (status-name) and a status type (status-type). The name of a state can be anything you want and you can specify this via the admin area of OTRS in the section for states. The state types can not be changed via the web interface, the database has to be modified if you'd want to add new types or if the name of types has to be changed. 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 following settings have to be modified to ensure that OTRS works with the changed state for "new":

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

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

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

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

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

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

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

mysql> quit
Bye
linux:~#
```

Now it is possible to use the new state type 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 status for new phone articles.

Ticket -> Frontend::Agent::Ticket::ViewPhoneOutbound > AgentTicketPhoneOutbound###StateType - to define the available next statuses for new phone articles you can here define the status types.

Ticket:Frontend::Agent::Ticket::ViewMove:Ticket::DefaultNextMoveStateType - to define the available next statuses in the AgentTicketMove screen you can here define the status types.

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 statuses in the bounce screen you can here define the status types.

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

Ticket -> Frontend::Agent::Ticket::ViewBulk > StateType - to define the available next statuses in the bulk action screen you can here define the status types.

Ticket -> Frontend::Agent::Ticket::ViewClose > StateDefault - to define the default next status in a close action.

Ticket -> Frontend::Agent::Ticket::ViewClose > StateType - to define the available next statuses in the close screen you can here define the status types.

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

Ticket -> Frontend::Agent::Ticket::ViewCompose > StateType - to define the available next statuses in the Compose (reply) screen you can here define the status types.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateDefault - to define the default next status in a Forward action.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateType - to define the available next statuses in the Forward screen you can here define the status types.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateDefault - to define the default next status in a Forward action.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateType - to define the available next statuses in the Forward screen you can here define the status types.

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

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

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

Ticket -> Core::Ticket > ViewableStateType - to define the status types that are displayed at various places in the system, for instance 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 been expired.

Chapter 13. Modifying ticket priorities

OTRS comes by default with five priority levels. If you'd like to customize this you can do so 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. Please take this into account when customizing priorities.

Due to the nature of OTRS it is not possible to remove existing priorities, you can set them to "Invalid" though.

Important

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

- Ticket:Core::Postmaster::PostmasterDefaultPriority - defines the default priority for all incoming email.
- 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] , especially 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 the database has to be changed manually and the new theme has to be added to the theme table. Below is an example of how to do this with MySQL:

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

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

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

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

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

Warning

Do not change the theme files shipped with OTRS, 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. The first step is to install GnuPG via the package manager of your operating system.
2. In the next step GnuPG has to be configured for the usage with OTRS. The needed directories for GnuPG and a private key have to be created. The following command has to be executed as OTRS user from a shell.

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

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

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

```
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
```

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: 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:~$
```

Like shown in the screen above for the most settings the default is OK. Only the values for the key owner have to be entered correctly and a proper password has to be specified for the key.

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

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

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

Via the last config option it is possible to specify the pairs of key IDs and their passwords for the own private keys. Because communication partners from outside write to the ticket system and their messages are encrypted with 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 following command is executed as OTRS user:

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

linux:~$
```

The ID of the private key can be found in the line that starts with "sub". It is a 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. The last step is the import of a customers public key. This ensures that also encrypted messages can be sent out to this customer. There are two possibilities to import a public key of a customer.

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

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

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

Chapter 17. S/MIME

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

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

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

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

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

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

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

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

```
-----
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:OTRS-state
Locality Name (eg, city) []:OTRS-town
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Your
company
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) []:OTRS Admin
Email Address []:otrs@your-domain.tld
otrs@linux:~/tmp> ls -la demoCA/
total 8
-rw-r--r--  1 otrs otrs 1330 2006-01-08 17:54 cacert.pem
drwxr-xr-x  2 otrs otrs  48 2006-01-08 17:53 certs
drwxr-xr-x  2 otrs otrs  48 2006-01-08 17:53 crl
-rw-r--r--  1 otrs otrs   0 2006-01-08 17:53 index.txt
drwxr-xr-x  2 otrs otrs  48 2006-01-08 17:53 newcerts
drwxr-xr-x  2 otrs otrs  80 2006-01-08 17:54 private
-rw-r--r--  1 otrs otrs  17 2006-01-08 17:54 serial
otrs@linux:~/tmp>
```

2. Generate a certificate request.

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

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

```
Request (and private key) is in newreq.pem
otrs@linux:~/tmp> ls -la
total 4
drwxr-xr-x  6 otrs otrs  232 2006-01-08 17:54 demoCA
-rw-r--r--  1 otrs otrs 1708 2006-01-08 18:04 newreq.pem
otrs@linux:~/tmp>
```

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

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

01:D9:1E:58:C0:6D:BF:27:ED:37:34:14:D6:04:AC:C4:64:98:7A:22
        X509v3 Authority Key Identifier:

keyid:10:4D:8D:4C:93:FD:2C:AA:9A:B3:26:80:6B:F5:D5:31:E2:8E:DB:A8
        DirName:/C=DE/ST=OTRS-state/L=OTRS-town/O=Your
Company/
        CN=OTRS admin/emailAddress=otrs@your-domain.tld
        serial:FD:85:F6:9F:14:07:16:C7

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

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Signed certificate is in newcert.pem
```

```
otrs@linux:~/tmp>
```

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

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

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

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

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

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

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

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

Chapter 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 rudimentary 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. 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'],
        },
    },
};

```

Please find below 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'],
    },
},
};
```

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,
- the user permission settings of the respective installation should be considered,
- it should be able to generate complex statistics,
- configuration of stats should be easy and quick,
- performance should be high,
- and a number of output variants should be supported.

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

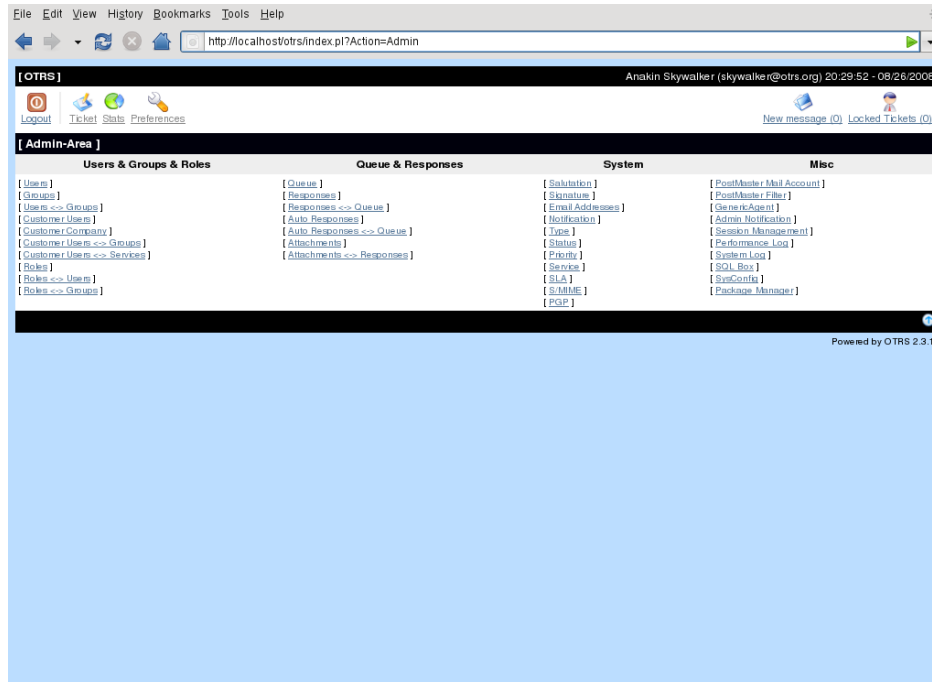
Statistical elements, i.e. files which supplement the functionalities of the stats module for specific requirements, can be integrated to model complex stats.

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.

All things considered, the stats module enables OTRS users a quick and easy evaluation of the activities on their system.

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 "Stats".



After the link has been activated, various action links for the stats module are shown in the navigation bar.

- "Overview"

is also shown automatically if the module link "Stats" is chosen

- "New"

rw rights required

- "Import"

rw rights required

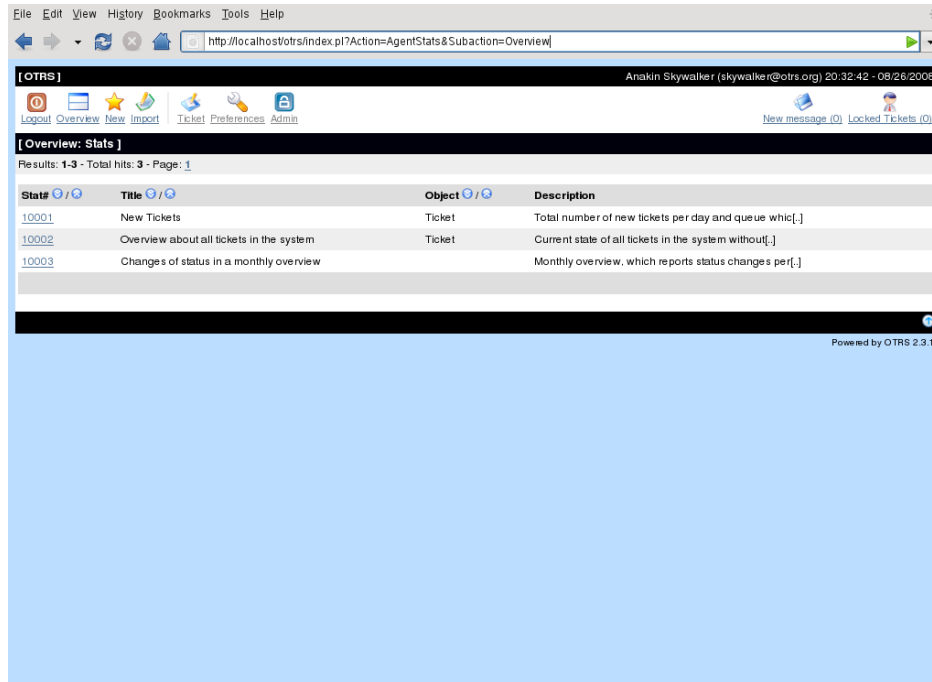
- "Preferences"

OTRS user interface for personal settings

Overview

A list of all pre-configured stats the agent can use.

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



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

- "Stat#"

the unique stat number

- "Title"

the stat title

- "Object"

Object used for generating the stat. In the case of a static stat no object is displayed as no dynamic object is used for the generation.

- "Description"

Excerpt of the description

A results display can be found above the list. It 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 according to their needs with the blue "up" and "down" icons in the headline.

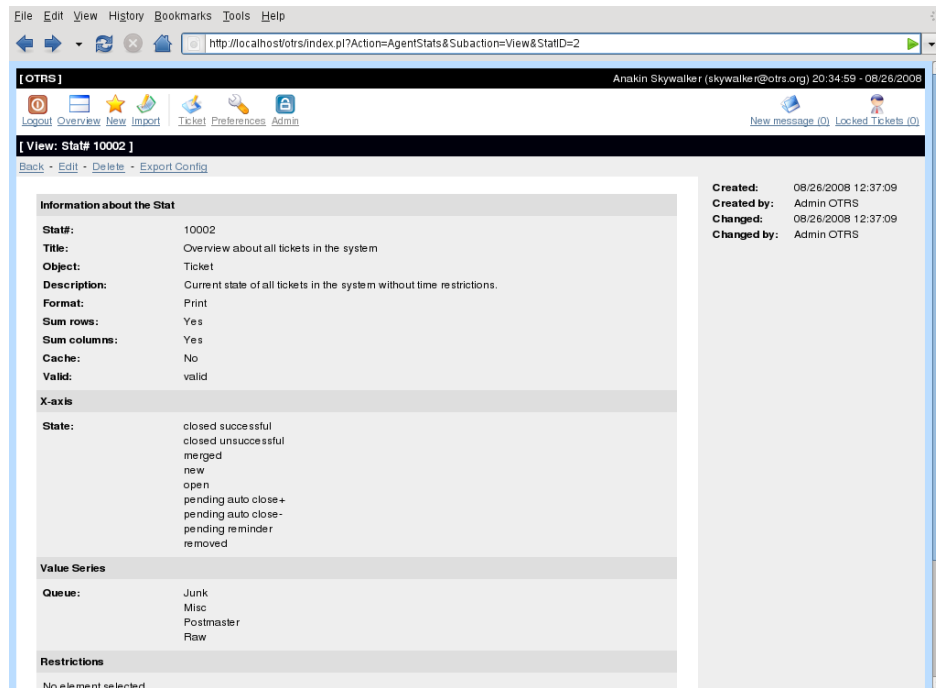
To generate a stat the right one is chosen by clicking the link with the stat number which leads to the "View" user interface.

A few sample stats are imported when the stats module is used for the first time.

Generate and view stats

The view user interface is supposed to provide the agent with an overview of the stat's pre-configuration.

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 cannot effect any changes or can change all attributes, respectively.



The page shows the following:

- the menu links

- "Back"

Link back to the starting point

- "Edit"

Link to the edit user interface of the stat (rw rights required)

- "Delete"

Delete a 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.

- Meta data (grey block on the right)

Provides information about the stat originators. Information is provided about who created the stat and when as well as who last changed it.

- The stat data itself

can be found in the central block of the user interface.

In the following, details to the stat input are provided as this input contains the core information.

General information is provided first.

- "Stat#"

The stat number

- "Title"

The stat title

- "Object"

Object used for generating the stat

- "Description"

The description provides more detailed information about the intention of the stat.

- "Format"

Stat output format. Depending on the configuration, following output formats can be chosen:

- "CSV" -> stat download in csv format
- "Print" -> printable html or pdf table
- Graph-lines
- "Graph-bars"
- "Graph-hbars"
- "Graph-points"
- "Graph-lines-points"
- "Graph-area"
- "Graph-pie"
- "Graphsize"

size which the graphic / chart may have

- indicated in pixels

this option is only given when the stat pre-configuration allows 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"

The "valid" element is set "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.

- "Exchange 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. Static stat generators can be integrated into the stat module for that reason. They were developed specifically for the particular evaluation. The various job definitions result in different views.

File Edit View History Bookmarks Tools Help

http://localhost/otrs/index.pl?Action=AgentStats&Subaction=View&StatID=3

[OTRS] Anakin Skywalker (skywalker@otrs.org) 20:36:37 - 08/26/2008

Logout Overview New Import Ticket Preferences Admin New message (0) Locked Tickets (0)

[View: Stat# 10003]

Back - Edit - Delete - Export Config

Information about the Stat

Stat#:	10003
Title:	Changes of status in a monthly overview
Object:	StateAction
Description:	Monthly overview, which reports status changes per day of a selected month.
Format:	Print
Sum rows:	Yes
Sum columns:	Yes
Cache:	Yes
Valid:	valid

Configurable params of static stat

Year:	2008
Month:	07

Start

Created: 08/26/2008 12:37:09
Created by: Admin OTRS
Changed: 08/26/2008 12:37:09
Changed by: Admin OTRS

[Note]
 With the input and select fields you can configure the stat at your needs. Which elements of a stat you can edit depends on your stats administrator who configured the stat.

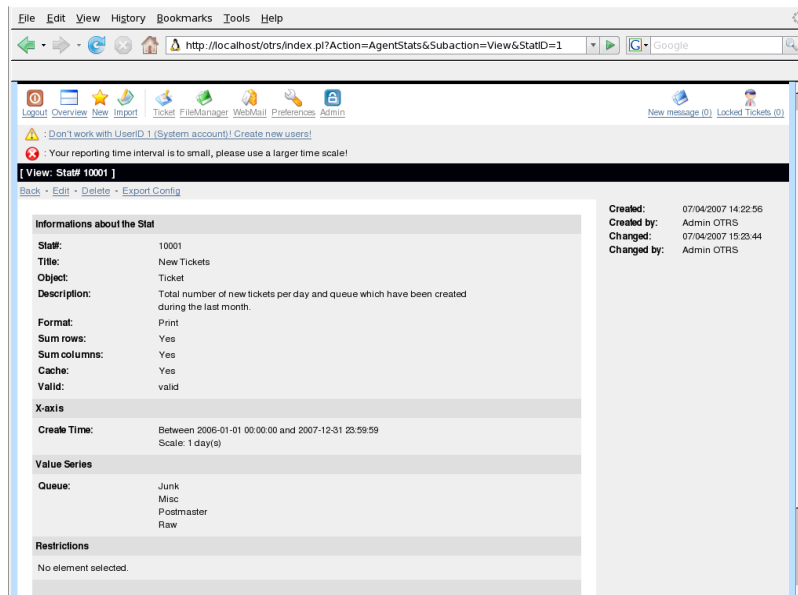
Powered by OTRS 2.3.1

- Dynamic stats can be displayed in two different ways:
 - Unchangeable settings

In this case, the originator of the stat has not allowed for alterations on the part of the agent who wants to generate a stat.

- changeable settings

The pre-configuration of such stats can be changed by the agent.



Pressing the "start" button at the bottom right end of the block is the last simple step to generate a stat. There are two possible reasons if the "start" button is not displayed:

1. The stat was set 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, the page is shown again after the "start" button was pushed and information about which input was incorrect is provided in the notification section.

Edit - New

Agents can call up the edit user interface of the stat module in two different ways:

1. via the "edit" menu link in the stat view to edit an existing stat configuration.
2. via the "new" link in the navigation bar.

These links are only visible to agents with write rights for this module.

The stats are edited with a wizard in four steps:

1. General information about the stat

2. Definition of the element for the x axis
3. Definition of the value series
4. Definition of the restrictions

Points 2-4 are only needed for the generation of dynamic stats. For a static stat only the entering of general information in 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 use interface is only displayed after the form has been filled out correctly.

1. "Common specification"

The first page of the wizard is the input user interface "Common specification". A great number of common specifications and settings can be edited on this page.

The screenshot shows the OTRS Admin interface for editing a common specification. The browser address bar shows the URL: `http://localhost/otrs/index.pl?Action=AgentStats&Subaction=EditSpecification&...`. The page title is "[OTRS] Admin OTRS (root@localhost) 13.05.10 - 07:18:2907". The navigation bar includes links like Logout, Overview, New, Import, Ticket, FileManager, WebMail, Preferences, and Admin. A warning message states: "Don't work with UserID 1 (System account)! Create new users!". The main content area is titled "[Edit: Stat# 10001]" and contains a form titled "Insert of the common specifications (1/4)". The form has four numbered steps: 1 (Title), 2 (Description), 3 (Dynamic-Object), and 4 (Permissions). The "Title" field is filled with "New Tickets". The "Description" field contains the text: "Total number of new tickets per day and queue which have been created during the last month.". The "Dynamic-Object" field is set to "Ticket". The "Permissions" field is a dropdown menu with "admin", "stats", and "users" options, where "stats" is selected. The "Format" field is a dropdown menu with "CSV", "Print", "graph-area", "graph-bars", and "graph-hbars" options, where "Print" is selected. The "Graphsize" field is a dropdown menu with "1200x800" and "1600x1200" options, where "1200x800" is selected.

- "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 selected at this step. The objects meet the requirements of the particular modules.
- "Static file"

A static file can be selected at this point. 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 configurations required.

- "Permission settings"

The 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 needing 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, the following output formats can be chosen:

- "CSV" -> stat output in csv format
- "Print" -> printable html or pdf table
- "graph-lines"
- "graph-bars"
- "graph-hbars"
- "graph-points"
- "graph-lines-points"
- "graph-area"
- "graph-pie"

More than one format can be selected.

- "Graphsize"

Select which size the charts can have.

- indicated in pixels
- 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"

If "yes" is chosen in the pulldown menu, the stat is amended by a column whose cells display 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"

The generated stats are cached in the file system if "yes" is selected in the pulldown menu. This saves computing power and time if the stat is called up again.

This function should only be used if the stat's content is no longer changing. Thorough checking is required before the caching function is used.

Caching is automatically prevented in the following situations:

- if the stat contains no time designation values (new data is added continuously)
- if a time designation value points to the future

If a cached stat is edited, all cached data is deleted.

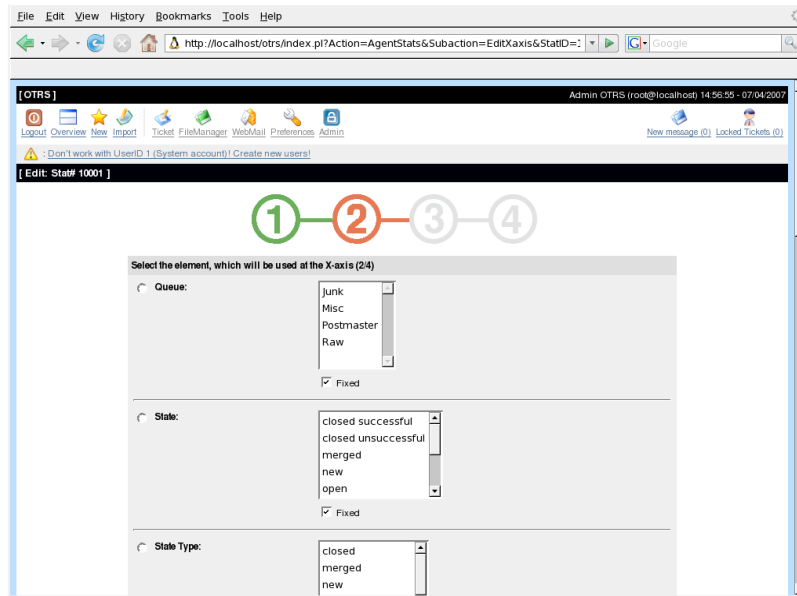
- "Valid"

The "valid" element 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.

By clicking the "Next" button the next user interface of the wizard is called up.

2. "xaxis"

The "Common statements" section is followed by the "xaxis" form.



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.

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.

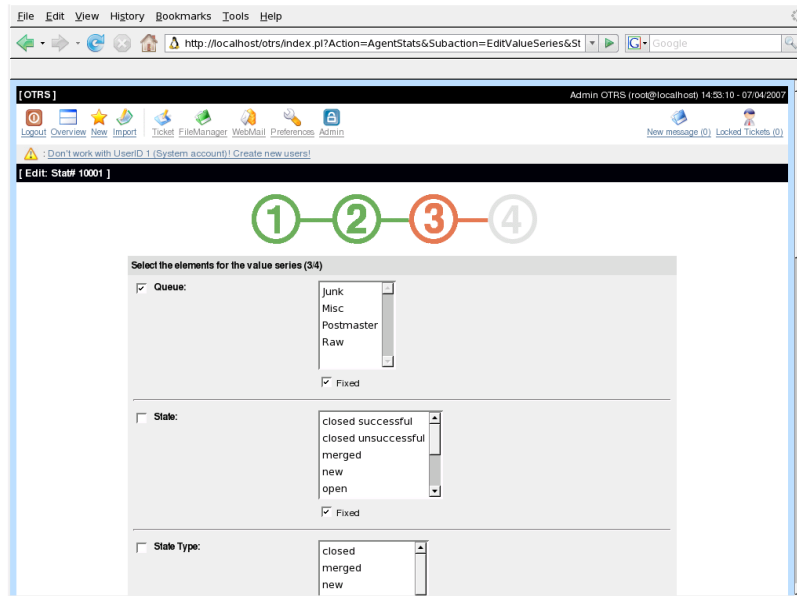
Time elements are different as time period and scale have to be stated.

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. "Value series"

In the third step of the stat pre-configuration the value series are defined. They will later on form the individual graphs of a diagram or the various series (tabular view).



If an element is selected, each chosen attribute corresponds to a value series.

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.

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

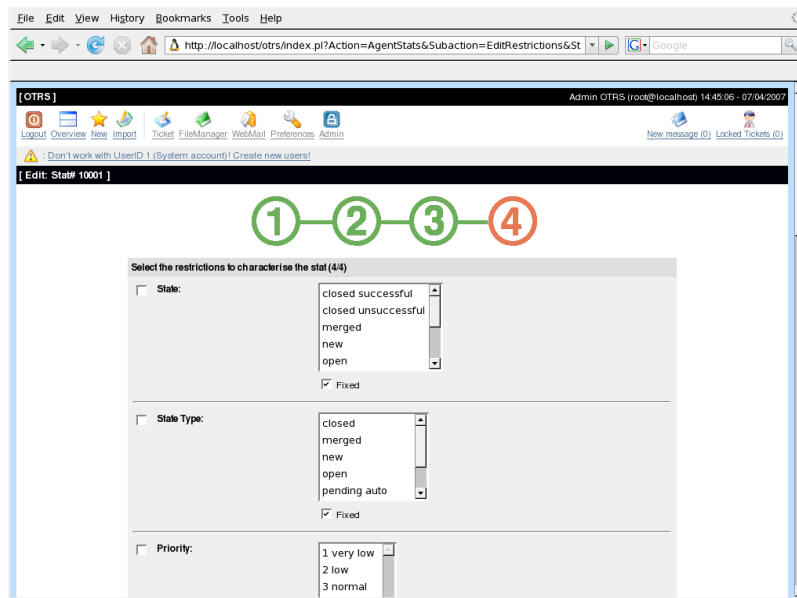
Selection of three or more elements does not make sense and is therefore prevented by an error message.

Additionally the same conditions apply to the selection of the attributes and the "Fixed" checkbox as to the "Xaxis" 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. "Restrictions"

The fourth and last step of the pre-configuration is the definition of restrictions. These restrictions facilitate to limit the results and are comparable to entering data into a search window.

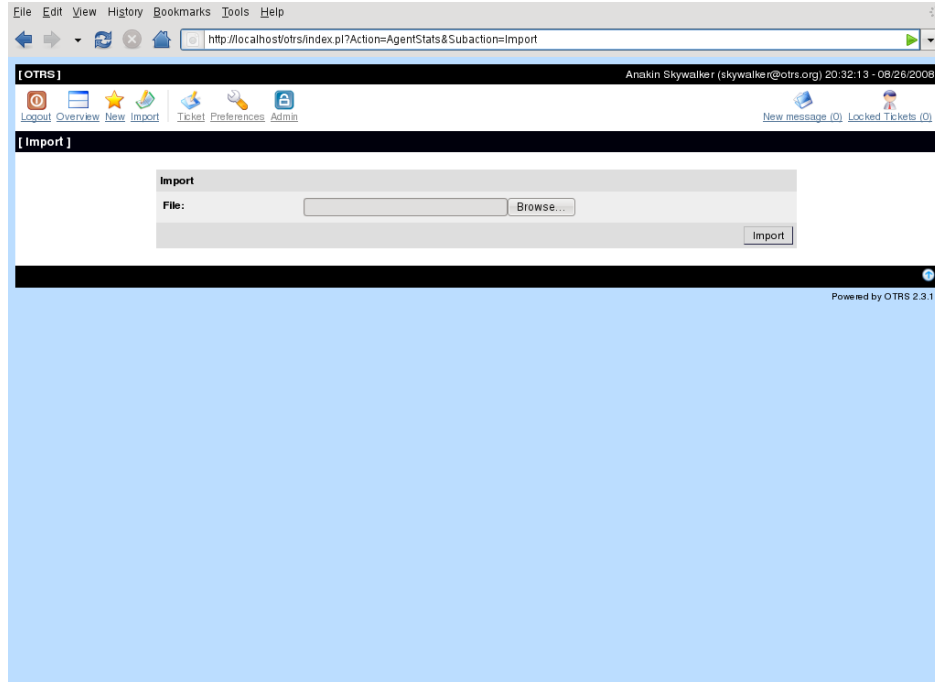


In quite a few cases, no restrictions at all must be set up.

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.

Import

The import user interface can be accessed by choosing "Import" in the navigation bar (rw rights required).



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 be imported into the production system afterwards.

The import is effected by an easy file upload. The "view" user interface of the imported stat is opened automatically afterwards.

Administration of the stat module by the OTRS administrator

This paragraph provides information about the tasks and possibilities of the OTRS administrator dealing with the stat module.

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" -> permission to configurate stats
- "ro" -> permission to generate pre-configured stats

The OTRS administrator decides whether agents with the permission to generate pre-configured stats are allocated ro rights in the "stats" group or whether their respective groups are added in the module registration in SysConfig.

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 stats module. Moreover, the configuration parameter `$Self->{'Frontend::Module'}->{'AgentStats'}` controls the arrangement and registration of the modules and icons within the stats module.

Administration of the stats module by the system administrator

Generally no system administrator is needed for the operation, configuration and maintenance. 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`).

Data base table

All stat configurations are implemented and administrated in XML. Therefore all stats configurations are stored in the database table "xml_storage". Other modules whose content is presented in xml format use this table as well.

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/mkStats.pl`

Caching

Whether the results of a stat are to be cached or not can be defined in the pre-configuration. Cached stat 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.

mkStats.pl

This file is saved in the `bin/` directory. It facilitates the generation of stats in the command line.

The following command line call is exemplary:

```
bin> perl mkStats.pl -n 10004 -o /output/dir
```

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 following call

```
bin> perl mkStats.pl --help
```

Automated stat generation - Cron

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 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.

Static stats

The stats module facilitates the generation of static stats. For every static stat a file exists in which the stat 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/`.

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;
```

Default stats

"It is not always necessary to reinvent the wheel..."

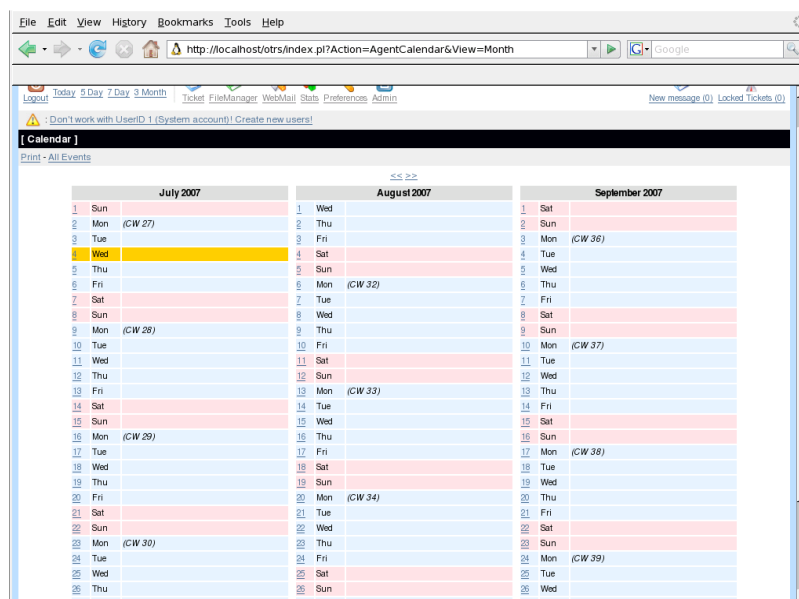
The stats module provides various default stats. Stats which are interesting for all OTRS users will in future be added to the default stats of the stats module package. Default stats are saved in the stats module xml format in the `scripts/test/sample/` directory.

Chapter 20. Additional applications

You can install additional application packages to extend the functionality of the OTRS framework. You can install packages via the package manager in the adminarea. The package manager 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.

Calendar

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

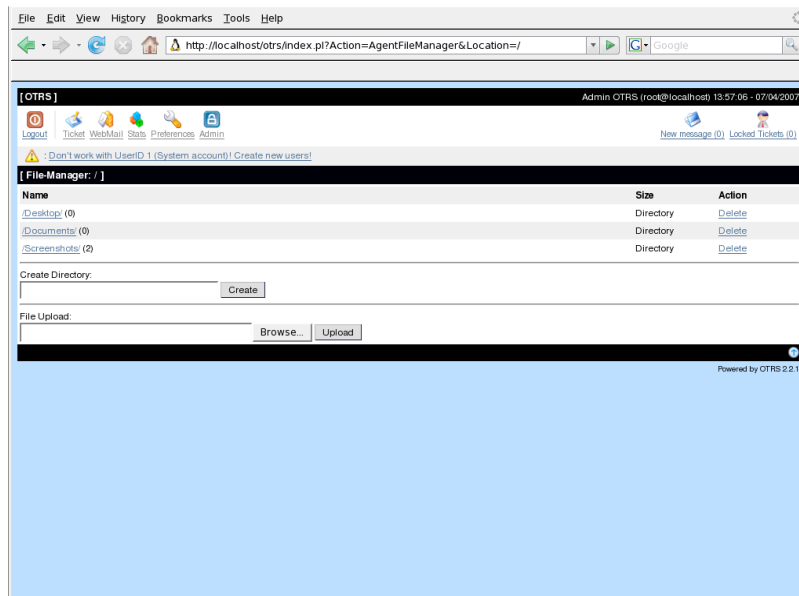


Content manager

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

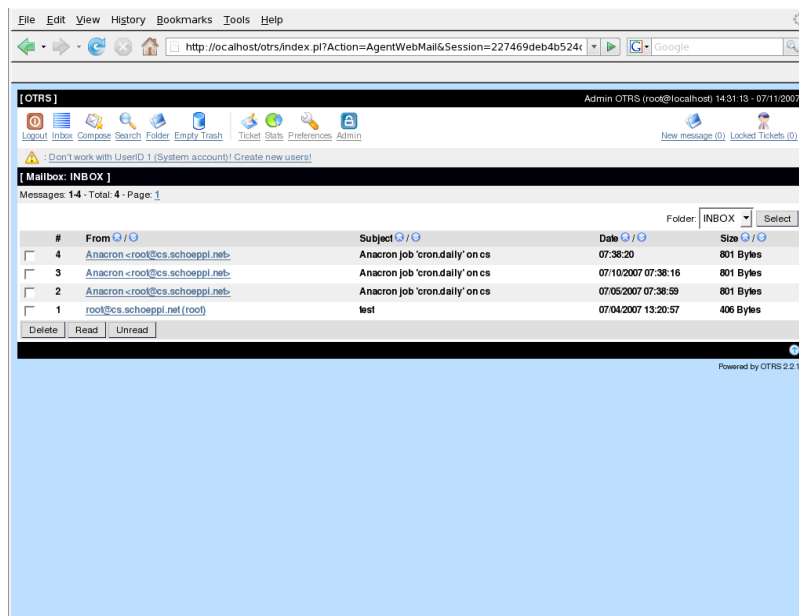
File manager

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



Web mailer

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

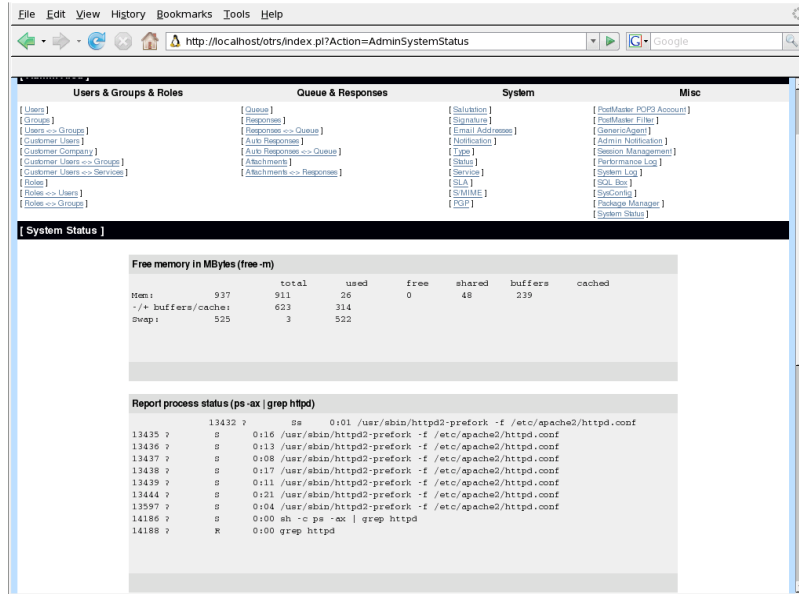


FAQ

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. Articles can be structured in groups and have different languages.

System status

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



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.

OTRS

There are several options to improve the performance of OTRS.

TicketIndexModule

You have two backend modules for the ticket index.

Kernel/Config.pm

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

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

TicketStorageModule

There are two different backend modules for the ticket/article storage.

Kernel/Config.pm

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

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

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

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

- Kernel::System::Ticket::ArticleStorageFS, store attachments etc. on the local file system. Note: Use it for larger set ups.

Pro: Faster!

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

Note: you can switch from one to the other backend on the fly.

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.

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.

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

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 and ask questions there. The folks on the PostgreSQL list are very friendly and can probably help <http://www.postgresql.org/community/lists/>.

Webserver

Of course you should use mod_perl 2.0 (<http://perl.apache.org/> [<http://perl.apache.org/>]). It's much faster (~ * 100) than pure cgi. But needs more RAM. So your httpd with mod_perl will be about 16 MB (~10MB shared).

Pre-establish database connections

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

Preloaded modules - startup.pl

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

Reload Perl modules when updated on disk

Per default Apache::Reload (mod_perl 2.0) is used in scripts/apache2-httpd.include.conf. Disable it and you will get 8% more speed. But remember to restart the web server if you change Kernel/Config.pm or any other modules! Important: this would also mean you can't use the OTRS-Packet-Manager via the web interface, you need to use CMD - bin/opm.pl.

Choosing the Right Strategy

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

mod_gzip/mod_deflate

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

Chapter 22. Backing up the system

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

Backup

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

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

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

Execute the following command to create a backup:

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

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

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

Restore

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

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

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


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

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

Appendix A. Additional Resources

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

Homepage OTRS.org

You can find our project homepage at <http://www.otrs.org/> [<http://www.otrs.org/>].



Mailing lists

The following table shows you our different community mailing lists.

Table A.1. Mailinglists

Name	Description	Homepage
announce at otrs.org	Low traffic list for announcements of new OTRS releases and security issues.	http://lists.otrs.org/cgi-bin/listinfo/announce [http://lists.otrs.org/cgi-bin/listinfo/announce]
otrs at otrs.org	Medium to high traffic list 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 at otrs.org	Medium to high traffic list where you can find all sorts of questions on, and support	http://lists.otrs.org/cgi-bin/listinfo/otrs-de [http://lists.otrs.org/cgi-bin/listinfo/otrs-de]

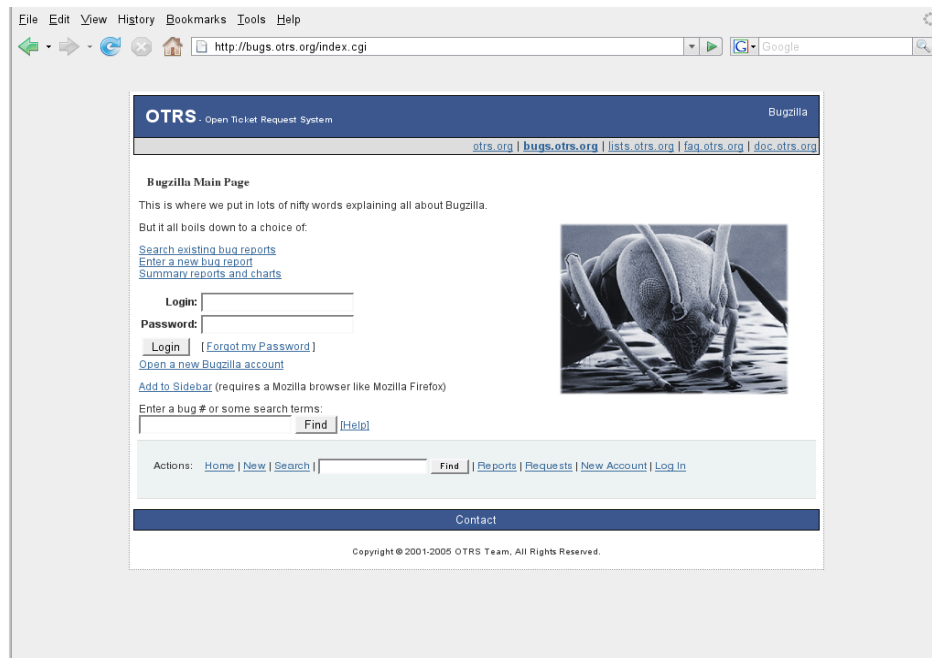
Name	Description	Homepage
	for, the product. This list is in German.	
dev at otrs.org	Medium traffic list. The OTRS developers discuss various design and implementation issues here.	http://lists.otrs.org/cgi-bin/listinfo/dev [http://lists.otrs.org/cgi-bin/listinfo/dev]
doc-de at otrs.org	Low traffic list with all sorts of questions, in German, 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 at otrs.org	Low traffic list for internationalization and localization questions. If you are or want to become a translator of the OTRS project or have any problems with one of our applications in an international environment, this is the right place.	http://lists.otrs.org/cgi-bin/listinfo/i18n [http://lists.otrs.org/cgi-bin/listinfo/i18n]
cvs-log at otrs.org	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/>].

Bug tracking

To submit bugs visit <http://bugs.otrs.org/> [<http://bugs.otrs.org/>] . Please take note of the difference between a bug and a configuration issue. Configuration issues are issues 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.

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!

Commercial Support

For services (support, consulting, development, and training) you can contact the company behind OTRS, OTRS AG. They have offices both in the US and in Germany. Look at their website for contact information: <http://www.otrs.com/en/corporate-navigation/contact/> [<http://www.otrs.com/en/corporate-navigation/contact/>]

Appendix B. Configuration Options Reference

Framework

Core

SecureMode

Description	Value
Description:	If enabled, the web installer (http://yourhost.example.com/otrs/installer.pl) can't be used. This setting should be set to "Yes" to prevent your system from being hijacked. If set to "No", you can reinstall your system. Your current basic configuration will be used to pre-populate the questions within the installer script. Also disables if not active, the GenericAgent, PackageManager and SQL Box, since it's possible to use it for destructive write 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>

ProductName

Description	Value
Description:	This setting controls the name of the application as is shown in the web interface as well as the tabs and title bar of your web browser.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'ProductName'} = 'OTRS';</code>

SystemID

Description	Value
Description:	The system identifier. Every ticket number contains this ID, and the session ID, of a http session string, starts with this number. This ensures that only tickets which belong to your system will be processed as follow-ups. This could be use full when communicating between two

Description	Value
	instances of OTRS. So be sure of what you choose when you set this setting.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SystemID'} = '10';</code>

FQDN

Description	Value
Description:	Please enter the fully qualified domain name of your system here. 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>

HttpType

Description	Value
Description:	Please enter the type of protocol, used by your web server, to serve the application. If you want to use the https protocol, instead of plain http, please specify it here. This has no affect on your web server's settings or behavior. This setting will not change the method of access to the application. If this setting 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>

ScriptAlias

Description	Value
Description:	Please enter the prefix to the scripts folder on your server as configured on your web server. This setting is used as a variable,

Description	Value
	OTRS_CONFIG_ScriptAlias 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->{'ScriptAlias'} = 'otrs/';</code>

AdminEmail

Description	Value
Description:	Please enter the system administrator's e-mail address. This is the e-mail address that will be displayed in error screens of the application.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'AdminEmail'} = 'admin@example.com';</code>

Organization

Description	Value
Description:	If configured, all e-mails sent by the application will contain a X-Header with this organization or company name.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Organization'} = 'Example Company';</code>

DefaultCharset

Description	Value
Description:	Please configure the default charset for the web interface to use. The charset "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 e-mails, or text, otherwise this could lead to problems. This setting should represent that of the charset used to create the database or, in some cases, that of the database management system being used.
Group:	Framework
SubGroup:	Core

Description	Value
	<pre> 'en_CA' => 'English (Canada)', 'en_GB' => 'English (United Kingdom)', 'es' => 'Espa&ntilde;ol', 'et' => 'Eesti', 'fa' => 'Persian (&#x0641;&#x0627;&#x0631;&#x0633;&#x0649;)', 'fi' => 'Suomi', 'fr' => 'Fran&ccedil;ais', 'hu' => 'Magyar', 'it' => 'Italiano', 'lv' => 'Latvijas', 'nb_NO' => 'Norsk bokm&aring;l', 'nl' => 'Nederlands', 'pl' => 'Polski', 'pt' => 'Portugu&ecirc;s', 'pt_BR' => 'Portugu&ecirc;s Brasileiro', 'ru' => 'Russian (&#x0420;&#x0443;&#x0441;&#x0441;&#x043a;&#x0438;&#x0439;)', 'sk_SK' => 'Slovak (Sloven&#x010d;ina)', 'sv' => 'Svenska', 'tr' => 'T&uuml;rk&ccedil;e', 'uk' => 'Ukrainian (&#x0423;&#x043a;&#x0440;&#x0430;&#x0457;&#x043d;&#x0441;&#x0444;)', 'vi_VN' => 'Vietnam (Vi&#x0246;t Nam)', 'zh_CN' => 'Chinese (Sim.) (&#x7b80;&#x4f53;&#x4e2d;&#x6587;)', 'zh_TW' => 'Chinese (Tradi.) (&#x6b63;&#x9ad4;&#x4e2d;&#x6587;)' }; </pre>

DefaultTheme

Description	Value
Description:	Please enter the default front-end (HTML) theme to be used by your agents and customers. The default themes are Standard and Lite. If you like, you can add your own theme. Please refer the the administrator manual located at http://doc.otrs.org/ .
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'DefaultTheme'} = 'Standard';</code>

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

Description	Value
	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>

CheckMXRecord

Description	Value
Description:	Should the application check the MX record of e-mail addresses before sending an e-mail or submitting a telephone or e-mail ticket?
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CheckMXRecord'} = '1';</pre>

CheckMXRecord::Nameserver

Description	Value
Description:	If it is necessary for the application to use a dedicated DNS server for the "CheckMXRecord" look-ups, please configure this setting here.
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'CheckMXRecord::Nameserver'} = 'ns.example.com';</pre>

CheckEmailAddresses

Description	Value
Description:	Should the application check the syntax of e-mail addresses?
Group:	Framework

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

CheckEmailValidAddress

Description	Value
Description:	If "CheckEmailAddresses" is set to "yes", it may be necessary to exclude some addresses from this check. Please enter a regex in this field for e-mail 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>

CheckEmailInvalidAddress

Description	Value
Description:	The regex found in this setting can filter all e-mail addresses that should not be used in the application.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CheckEmailInvalidAddress'} = '@(aa aaa aaaa aaaaa abc any anywhere anonymous bar demo example foo hello hallo nospam nowhere null some somewhere test teste. there user xx xxx xxxx)\\. (.. ...) \$';</code>

CGILogPrefix

Description	Value
Description:	Please enter the text that should appear in the log file to denote a CGI script entry.
Group:	Framework
SubGroup:	Core

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

DemoSystem

Description	Value
Description:	Should the system be run 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>

SwitchToUser

Description	Value
Description:	Should the administrators be allowed, via the users administration panel, to login as other users?
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SwitchToUser'} = '0';</code>

NotificationSenderName

Description	Value
Description:	Please enter 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:	

Description	Value
	<pre>\$Self->{'NotificationSenderName'} = 'OTRS Notification Master';</pre>

NotificationSenderEmail

Description	Value
Description:	Please enter the e-mail address that should be used by the application when sending notifications. The e-mail 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::Agnet::Move.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'NotificationSenderEmail'} = 'otrs@<OTRS_CONFIG_FQDN>';</pre>

System::Customer::Permission

Description	Value
Description:	These are 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
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'System::Customer::Permission'} = ['ro', 'rw'];</pre>

LanguageDebug

Description	Value
Description:	Should debugging be turned to debug your 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

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

Secure::DisableBanner

Description	Value
Description:	If enabled, the OTRS version tag will be removed from the HTML headers.
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Secure::DisableBanner'} = '0';</code>

Core::LinkObject

LinkObject::ViewMode

Description	Value
Description:	Determines the way the linked objects are displayed in each zoom mask.
Group:	Framework
SubGroup:	Core::LinkObject
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LinkObject::ViewMode'} = 'Simple';</code>

LinkObject::Type###Normal

Description	Value
Description:	This setting 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. If the values are different, the resulting link is a directional link.
Group:	Framework
SubGroup:	Core::LinkObject
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	<pre>\$Self->{'LinkObject::Type'}->{'Normal'} = { 'SourceName' => 'Normal', 'TargetName' => 'Normal' };</pre>

LinkObject::Type###ParentChild

Description	Value
Description:	This setting 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. If the values are different, the resulting link is a directional link.
Group:	Framework
SubGroup:	Core::LinkObject
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'LinkObject::Type'}->{'ParentChild'} = { 'SourceName' => 'Parent', 'TargetName' => 'Child' };</pre>

LinkObject::TypeGroup###0001

Description	Value
Description:	This setting 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.
Group:	Framework
SubGroup:	Core::LinkObject
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'LinkObject::TypeGroup'}->{'0001'} = ['Normal', 'ParentChild'];</pre>

LinkObject::PossibleLink###0200

Description	Value
Description:	This setting defines that a 'Ticket' object can be linked with other 'Ticket' objects using the 'Normal' link type.
Group:	Framework
SubGroup:	Core::LinkObject

Description	Value
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'LinkObject::PossibleLink'}->{'0200'} = { 'Object1' => 'Ticket', 'Object2' => 'Ticket', 'Type' => 'Normal' };</pre>

LinkObject::PossibleLink###0201

Description	Value
Description:	This setting defines that a 'Ticket' object can be linked with other 'Ticket' objects using the 'ParentChild' link type.
Group:	Framework
SubGroup:	Core::LinkObject
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'LinkObject::PossibleLink'}->{'0201'} = { 'Object1' => 'Ticket', 'Object2' => 'Ticket', 'Type' => 'ParentChild' };</pre>

Core::Log

LogModule

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

LogModule::SysLog::Facility

Description	Value
Description:	If "SysLog" was selected for LogModule, you can specify a special log facility.

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

LogModule::SysLog::LogSock

Description	Value
Description:	If "SysLog" was selected for LogModule, you can specify a special log sock (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>

LogModule::SysLog::Charset

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

LogModule::LogFile

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

LogModule::LogFile::Date

Description	Value
Description:	Set this config parameter to "Yes", if you want to add a suffix with the actual year and month to the otrs logfile. A logfile for every month will be created.
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LogModule::LogFile::Date'} = '0';</code>

Core::MIME-Viewer

MIME-Viewer###application/excel

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

MIME-Viewer###application/msword

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

MIME-Viewer###application/pdf

Description	Value
Description:	Path to programm to view PDF Documents in the web interface.
Group:	Framework

Description	Value
SubGroup:	Core::MIME-Viewer
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'MIME-Viewer'}->{'application/pdf'} = 'pdftohtml -stdout -i';</pre>

MIME-Viewer###text/xml

Description	Value
Description:	Path to converter to view XML files in the web interface.
Group:	Framework
SubGroup:	Core::MIME-Viewer
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'MIME-Viewer'}->{'text/xml'} = '<OTRS_CONFIG_Home>/scripts/tools/xml2html.pl';</pre>

Core::MirrorDB

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:	<pre>\$Self->{'Core::MirrorDB::DSN'} = 'DBI:mysql:database=mirrordb;host=mirrordbhost';</pre>

Core::MirrorDB::User

Description	Value
Description:	If you want to use a mirror database for agent ticket fulltext search or to generate stats, you can specify the user to authenticate to this database.
Group:	Framework
SubGroup:	Core::MirrorDB
Valid:	0

Description	Value
Required:	0
Config-Setting:	<code>\$Self->{'Core::MirrorDB::User'} = 'some_user';</code>

Core::MirrorDB::Password

Description	Value
Description:	If you want to use a mirror database for agent ticket fulltext search or to generate stats, you can specify the password to authenticate to this database.
Group:	Framework
SubGroup:	Core::MirrorDB
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Core::MirrorDB::Password'} = 'some_password';</code>

Core::PDF

PDF

Description	Value
Description:	Enable 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>

PDF::LogoFile

Description	Value
Description:	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>

PDF::PageSize

Description	Value
Description:	Standard-size of PDF pages.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PDF::PageSize'} = 'a4';</code>

PDF::MaxPages

Description	Value
Description:	Maximum number of pages per PDF file.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PDF::MaxPages'} = '100';</code>

PDF::TTFontFile###Proportional

Description	Value
Description:	Path and TTF-File to handle proportional font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PDF::TTFontFile'}->{'Proportional'} = 'DejaVuSans.ttf';</code>

PDF::TTFontFile###ProportionalBold

Description	Value
Description:	Path and TTF-File to handle bold proportional font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	<code>\$Self->{'PDF::TTFontFile'}->{'ProportionalBold'} = 'DejaVuSans-Bold.ttf';</code>

PDF::TTFontFile###ProportionalItalic

Description	Value
Description:	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>

PDF::TTFontFile###ProportionalBoldItalic

Description	Value
Description:	Path and TTF-File to handle bold italic proportional font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PDF::TTFontFile'}->{'ProportionalBoldItalic'} = 'DejaVuSans-BoldOblique.ttf';</code>

PDF::TTFontFile###Monospaced

Description	Value
Description:	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>

PDF::TTFontFile###MonospacedBold

Description	Value
Description:	Path and TTF-File to handle bold monospaced font in PDF documents.

Description	Value
Group:	Framework
SubGroup:	Core::PDF
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'PDF::TTFontFile'}->{'MonospacedBold'} = 'DejaVuSansMono-Bold.ttf';</code>

PDF::TTFontFile###MonospacedItalic

Description	Value
Description:	Path and TTF-File to handle italic monospaced font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'PDF::TTFontFile'}->{'MonospacedItalic'} = 'DejaVuSansMono-Oblique.ttf';</code>

PDF::TTFontFile###MonospacedBoldItalic

Description	Value
Description:	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-BoldOblique.ttf';</code>

Core::Package

Package::FileUpload

Description	Value
Description:	Enable file upload in package manager frontend?
Group:	Framework
SubGroup:	Core::Package
Valid:	1

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

Package::RepositoryRoot

Description	Value
Description:	Location to get online repository list for additional packages. The first available result will be used.
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Package::RepositoryRoot'} = ['http://ftp.otrs.org/pub/otrs/misc/packages/ repository.xml', 'http://otrs.org/repository.xml'];</code>

Package::RepositoryList

Description	Value
Description:	List of online repositories (for example you also can use other installations as repository by using 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:	<code>\$Self->{'Package::RepositoryList'} = { 'ftp://ftp.example.com/pub/otrs/misc/packages/' => '[Example] ftp://ftp.example.com/' };</code>

Package::RepositoryAccessRegExp

Description	Value
Description:	IP RegExp for accessing local repository. You need to enable this to have access to your local repository. Package::RepositoryList is required on the remote host.
Group:	Framework
SubGroup:	Core::Package

Description	Value
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Package::RepositoryAccessRegExp'} = '127\\.0\\.0\\.1';</code>

Package::Timeout

Description	Value
Description:	Timeout for package downloads (in seconds). Overwrites "WebUserAgent::Timeout".
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Package::Timeout'} = '15';</code>

Package::Proxy

Description	Value
Description:	Fetch 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>

Core::PerformanceLog

PerformanceLog

Description	Value
Description:	Enable PerformanceLog to log the page response time. Take care, this will take system performance (you also need to enable Frontend::Module###AdminPerformanceLog!).
Group:	Framework
SubGroup:	Core::PerformanceLog
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	<code>\$Self->{'PerformanceLog'} = '0';</code>

PerformanceLog::File

Description	Value
Description:	Logfile of PerformanceLog.
Group:	Framework
SubGroup:	Core::PerformanceLog
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PerformanceLog::File'} = '<OTRS_CONFIG_Home>/var/log/Performance.log';</code>

PerformanceLog::FileMax

Description	Value
Description:	Max. size in MB of log file.
Group:	Framework
SubGroup:	Core::PerformanceLog
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PerformanceLog::FileMax'} = '25';</code>

Core::SOAP

SOAP::User

Description	Value
Description:	The user name 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>

SOAP::Password

Description	Value
Description:	The password to access the SOAP handle (bin/cgi-bin/rpc.pl).
Group:	Framework

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

Core::Sendmail

SendmailModule

Description	Value
Description:	Module to send emails. "Sendmail" directly uses the sendmail binary of your operating system. "SMTP/SMTPS" uses a specified (external) mailserver. "DoNotSendEmail" is not sending emails and useful for test systems.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SendmailModule'} = 'Kernel::System::Email::Sendmail';</code>

SendmailModule::CMD

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

SendmailModule::Host

Description	Value
Description:	If "SMTP/SMTPS" was selected for SendmailModule you have to specify the mailhost that sends out the mails.
Group:	Framework
SubGroup:	Core::Sendmail

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SendmailModule::Host'} = 'mail.example.com';</code>

SendmailModule::Port

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

SendmailModule::AuthUser

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

SendmailModule::AuthPassword

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

SendmailBcc

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

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>

SendmailEncodingForce

Description	Value
Description:	To force encoding of outgoing emails (7bit 8bit quoted-printable base64).
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'SendmailEncodingForce'} = 'base64';</code>

Core::Session

SessionModule

Description	Value
Description:	Module to store the session data. Advantage of "DB" is that you can split the frontend server from the db server. "FS" or "IPC" is much faster.

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

SessionName

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

SessionCheckRemoteIP

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

SessionDeleteIfNotRemoteID

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

Description	Value
	<code>\$Self->{'SessionDeleteIfNotRemoteID'} = '1';</code>

SessionMaxTime

Description	Value
Description:	Maximal valid time for a session id (in seconds).
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionMaxTime'} = '57600';</code>

SessionMaxIdleTime

Description	Value
Description:	A session will be killed and the user will be logged out after this time of inactivity (in seconds).
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionMaxIdleTime'} = '21600';</code>

SessionDeleteIfTimeToOld

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

SessionUseCookie

Description	Value
Description:	Should the session management use html cookies? If html cookies are disabled or if the client browser disabled html cookies, then the system will work as usual and append the session id to the links.
Group:	Framework

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

SessionUseCookieAfterBrowserClose

Description	Value
Description:	Store cookies after the browser has been closed?
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'SessionUseCookieAfterBrowserClose'} = '0';</code>

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
Config-Setting:	<code>\$Self->{'SessionCSRFProtection'} = '1';</code>

SessionDir

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

SessionTable

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

SessionTableID

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

SessionTableValue

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

CustomerPanelSessionName

Description	Value
Description:	Name of the key for customer sessions.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'CustomerPanelSessionName'} = 'CSID';</code>

Core::SpellChecker

SpellChecker

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

SpellCheckerBin

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

SpellCheckerDictDefault

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

SpellCheckerIgnore

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

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

Core::Stats

Stats::StatsHook

Description	Value
Description:	To set the stats hook.
Group:	Framework
SubGroup:	Core::Stats
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Stats::StatsHook'} = 'Stat#';</pre>

Stats::StatsStartNumber

Description	Value
Description:	Start stats counting from e. g. 10000. Every new stat increments this number.
Group:	Framework
SubGroup:	Core::Stats
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Stats::StatsStartNumber'} = '10000';</pre>

Stats::MaxXaxisAttributes

Description	Value
Description:	Default maximum number of X-axis attributes for the time scale.
Group:	Framework

Description	Value
SubGroup:	Core::Stats
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Stats::MaxXaxisAttributes'} = '1000';</code>

Stats::DynamicObjectRegistration###Ticket

Description	Value
Description:	Here you can decide if the common stats module may generate stats about tickets.
Group:	Framework
SubGroup:	Core::Stats
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Stats::DynamicObjectRegistration'}->{'Ticket'} = { 'Module' => 'Kernel::System::Stats::Dynamic::Ticket' };</code>

Stats::DynamicObjectRegistration###TicketList

Description	Value
Description:	Here you can decide if the stats module may generate ticket lists.
Group:	Framework
SubGroup:	Core::Stats
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Stats::DynamicObjectRegistration'}->{'TicketList'} = { 'Module' => 'Kernel::System::Stats::Dynamic::TicketList' };</code>

Stats::DynamicObjectRegistration###TicketAccountedTime

Description	Value
Description:	Here you can decide if the stats module may generate stats about the accounted time for tickets.
Group:	Framework
SubGroup:	Core::Stats

Description	Value
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Stats::DynamicObjectRegistration'}- >{'TicketAccountedTime'} = { 'Module' => 'Kernel::System::Stats::Dynamic::TicketAccountedTime' };</pre>

Stats::DynamicObjectRegistration###TicketSolutionResponseTime

Description	Value
Description:	Here you can decide if the stats module may generate stats about the solution and response time for tickets.
Group:	Framework
SubGroup:	Core::Stats
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Stats::DynamicObjectRegistration'}- >{'TicketSolutionResponseTime'} = { 'Module' => 'Kernel::System::Stats::Dynamic::TicketSolutionResponseTime' };</pre>

Core::Stats::Graph

Stats::Graph::t_margin

Description	Value
Description:	Top margin of chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Stats::Graph::t_margin'} = '10';</pre>

Stats::Graph::l_margin

Description	Value
Description:	Left margin of chart.
Group:	Framework
SubGroup:	Core::Stats::Graph

Description	Value
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::l_margin'} = '10';</code>

Stats::Graph::b_margin

Description	Value
Description:	Bottom margin of chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::b_margin'} = '10';</code>

Stats::Graph::r_margin

Description	Value
Description:	Right margin of chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::r_margin'} = '20';</code>

Stats::Graph::bgclr

Description	Value
Description:	Backgroundcolor of the picture.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::bgclr'} = 'white';</code>

Stats::Graph::transparent

Description	Value
Description:	Is the picture transparent?
Group:	Framework

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

Stats::Graph::fgclr

Description	Value
Description:	Border color of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::fgclr'} = 'black';</code>

Stats::Graph::boxclr

Description	Value
Description:	Background color of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::boxclr'} = 'white';</code>

Stats::Graph::accentclr

Description	Value
Description:	Bordercolor of the legend.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::accentclr'} = 'black';</code>

Stats::Graph::legendclr

Description	Value
Description:	Text color of the legend.

Description	Value
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::legendclr'} = 'black';</code>

Stats::Graph::textclr

Description	Value
Description:	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>

Stats::Graph::dclrs

Description	Value
Description:	Colors for the graphs.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::dclrs'} = ['red', 'green', 'blue', 'yellow', 'black', 'purple', 'orange', 'pink', 'marine', 'cyan', 'lgray', 'lblue', 'lyellow', 'lgreen', 'lred', 'lpurple', 'lorange',]</code>

Description	Value
	<code>'lbrown'</code> <code>];</code>

Stats::Graph::line_width

Description	Value
Description:	Boldness of the line drawn by the graph.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::line_width'} = '1';</code>

Stats::Graph::legend_placement

Description	Value
Description:	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:	<code>\$Self->{'Stats::Graph::legend_placement'} = 'BC';</code>

Stats::Graph::legend_spacing

Description	Value
Description:	Spacing of the legends.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::legend_spacing'} = '4';</code>

Stats::Graph::legend_marker_width

Description	Value
Description:	Width of the legend.
Group:	Framework

Description	Value
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::Graph::legend_marker_width'} = '12';</code>

Stats::Graph::legend_marker_height

Description	Value
Description:	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>

Core::Time

TimeInputFormat

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

TimeCalendarLookup

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

TimeShowAlwaysLong

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

TimeZone

Description	Value
Description:	Set 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>

TimeZoneUser

Description	Value
Description:	Set 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>

TimeZoneUserBrowserAutoOffset

Description	Value
Description:	Set 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>

TimeVacationDays

Description	Value
Description:	Add your permanent vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	<pre>\$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' } };</pre>

TimeVacationDaysOneTime

Description	Value
Description:	One time vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeVacationDaysOneTime'} = { '2004' => { '1' => { '1' => 'test' } } };</pre>

TimeWorkingHours

Description	Value
Description:	Hours and week days to count the working time.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	<pre> \$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', </pre>

Description	Value
	<pre>'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>

Core::Time::Calendar1

TimeZone::Calendar1Name

Description	Value
Description:	Calendar Name.
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	<code>\$Self->{'TimeZone::Calendar1Name'} = 'Calendar Name 1';</code>

TimeZone::Calendar1

Description	Value
Description:	The time zone of the queue.
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeZone::Calendar1'} = '0';</code>

TimeVacationDays::Calendar1

Description	Value
Description:	Add your permanent vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'TimeVacationDays::Calendar1'} = { '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' } }; </pre>

TimeVacationDaysOneTime::Calendar1

Description	Value
Description:	One time vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar1

Description	Value
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeVacationDaysOneTime::Calendar1'} = { '2004' => { '1' => { '1' => 'test' } } };</pre>

TimeWorkingHours::Calendar1

Description	Value
Description:	Hours and week days to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1
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',</pre>

Description	Value
	<pre>'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>

Core::Time::Calendar2

TimeZone::Calendar2Name

Description	Value
Description:	Calendar Name.
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeZone::Calendar2Name'} = 'Calendar Name 2';</pre>

TimeZone::Calendar2

Description	Value
Description:	The time zone of the queue.
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeZone::Calendar2'} = '0';</pre>

TimeVacationDays::Calendar2

Description	Value
Description:	Add your permanent vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1
Config-Setting:	<pre>\$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' } }</pre>

Description	Value
	<pre> } }; </pre>

TimeVacationDaysOneTime::Calendar2

Description	Value
Description:	One time vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'TimeVacationDaysOneTime::Calendar2'} = { '2004' => { '1' => { '1' => 'test' } } } }; </pre>

TimeWorkingHours::Calendar2

Description	Value
Description:	Hours and week days to count the working time.
Group:	Framework
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', </pre>

Description	Value
	<pre> '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', </pre>

Description	Value
	<pre> '14', '15', '16', '17', '18', '19', '20'] }; </pre>

Core::Time::Calendar3

TimeZone::Calendar3Name

Description	Value
Description:	Calendar Name.
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1
Config-Setting:	<pre> \$self->{'TimeZone::Calendar3Name'} = 'Calendar Name 3'; </pre>

TimeZone::Calendar3

Description	Value
Description:	The time zone of the queue.
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1
Config-Setting:	<pre> \$self->{'TimeZone::Calendar3'} = '0'; </pre>

TimeVacationDays::Calendar3

Description	Value
Description:	Add your permanent vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	<pre>\$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' } };</pre>

TimeVacationDaysOneTime::Calendar3

Description	Value
Description:	One time vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeVacationDaysOneTime::Calendar3'} = { '2004' => { '1' => { '1' => 'test' } } };</pre>

TimeWorkingHours::Calendar3

Description	Value
Description:	Hours and week days 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',</pre>

Description	Value
	<pre> '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', </pre>

Description	Value
	<pre> '17', '18', '19', '20'], 'Wed' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'] }; </pre>

Core::Time::Calendar4

TimeZone::Calendar4Name

Description	Value
Description:	Calendar Name.
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'TimeZone::Calendar4Name'} = 'Calendar Name 4'; </pre>

TimeZone::Calendar4

Description	Value
Description:	The time zone of the queue.
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'TimeZone::Calendar4'} = '0'; </pre>

TimeVacationDays::Calendar4

Description	Value
Description:	Add your permanent vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1
Config-Setting:	<pre>\$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' } };</pre>

TimeVacationDaysOneTime::Calendar4

Description	Value
Description:	One time vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeVacationDaysOneTime::Calendar4'} = { '2004' => { '1' => { '1' => 'test' } } };</pre>

TimeWorkingHours::Calendar4

Description	Value
Description:	Hours and week days to count the working time.
Group:	Framework

Description	Value
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'] } </pre>

Description	Value
	<pre>], '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>

Core::Time::Calendar5

TimeZone::Calendar5Name

Description	Value
Description:	Calendar Name.
Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1
Config-Setting:	<pre> \$self->{'TimeZone::Calendar5Name'} = 'Calendar Name 5'; </pre>

TimeZone::Calendar5

Description	Value
Description:	The time zone of the queue.
Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeZone::Calendar5'} = '0';</code>

TimeVacationDays::Calendar5

Description	Value
Description:	Add your permanent vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1
Config-Setting:	<pre> \$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' } }; </pre>

TimeVacationDaysOneTime::Calendar5

Description	Value
Description:	One time vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDaysOneTime::Calendar5'} = {</code>

Description	Value
	<pre>'2004' => { '1' => { '1' => 'test' } } };</pre>

TimeWorkingHours::Calendar5

Description	Value
Description:	Hours and week days 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' => [</pre>

Description	Value
	<pre> '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>

Core::Time::Calendar6

TimeZone::Calendar6Name

Description	Value
Description:	Calendar Name.
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeZone::Calendar6Name'} = 'Calendar Name 6';</code>

TimeZone::Calendar6

Description	Value
Description:	The time zone of the queue.
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeZone::Calendar6'} = '0';</code>

TimeVacationDays::Calendar6

Description	Value
Description:	Add your permanent vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
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>

Description	Value
	<pre> } }; </pre>

TimeVacationDaysOneTime::Calendar6

Description	Value
Description:	One time vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'TimeVacationDaysOneTime::Calendar6'} = { '2004' => { '1' => { '1' => 'test' } } } }; </pre>

TimeWorkingHours::Calendar6

Description	Value
Description:	Hours and week days 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', </pre>

Description	Value
	<pre> '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', </pre>

Description	Value
	<pre> '14', '15', '16', '17', '18', '19', '20'] }; </pre>

Core::Time::Calendar7

TimeZone::Calendar7Name

Description	Value
Description:	Calendar Name.
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	<pre> \$self->{'TimeZone::Calendar7Name'} = 'Calendar Name 7'; </pre>

TimeZone::Calendar7

Description	Value
Description:	The time zone of the queue.
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	<pre> \$self->{'TimeZone::Calendar7'} = '0'; </pre>

TimeVacationDays::Calendar7

Description	Value
Description:	Add your permanent vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	<pre>\$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' } };</pre>

TimeVacationDaysOneTime::Calendar7

Description	Value
Description:	One time vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeVacationDaysOneTime::Calendar7'} = { '2004' => { '1' => { '1' => 'test' } } };</pre>

TimeWorkingHours::Calendar7

Description	Value
Description:	Hours and week days 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',</pre>

Description	Value
	<pre> '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', </pre>

Description	Value
	<pre> '17', '18', '19', '20'], 'Wed' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'] }; </pre>

Core::Time::Calendar8

TimeZone::Calendar8Name

Description	Value
Description:	Calendar Name.
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'TimeZone::Calendar8Name'} = 'Calendar Name 8'; </pre>

TimeZone::Calendar8

Description	Value
Description:	The time zone of the queue.
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'TimeZone::Calendar8'} = '0'; </pre>

TimeVacationDays::Calendar8

Description	Value
Description:	Add your permanent vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1
Config-Setting:	<pre>\$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' } };</pre>

TimeVacationDaysOneTime::Calendar8

Description	Value
Description:	One time vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TimeVacationDaysOneTime::Calendar8'} = { '2004' => { '1' => { '1' => 'test' } } };</pre>

TimeWorkingHours::Calendar8

Description	Value
Description:	Hours and week days to count the working time.
Group:	Framework

Description	Value
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'] } </pre>

Description	Value
	<pre>], '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>

Core::Time::Calendar9

TimeZone::Calendar9Name

Description	Value
Description:	Calendar Name.
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	<pre> \$self->{'TimeZone::Calendar9Name'} = 'Calendar Name 9'; </pre>

TimeZone::Calendar9

Description	Value
Description:	The time zone of the queue.
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeZone::Calendar9'} = '0';</code>

TimeVacationDays::Calendar9

Description	Value
Description:	Add your permanent vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	<pre> \$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' } }; </pre>

TimeVacationDaysOneTime::Calendar9

Description	Value
Description:	One time vacation days. Use single-digit pattern for months January to September and for days of month lower than ten.
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TimeVacationDaysOneTime::Calendar9'} = {</code>

Description	Value
	<pre>'2004' => { '1' => { '1' => 'test' } } };</pre>

TimeWorkingHours::Calendar9

Description	Value
Description:	Hours and week days 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' => [</pre>

Description	Value
	<pre> '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>

Core::Web

Frontend::WebPath

Description	Value
Description:	URL base path of icons, CSS and Java Script.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::WebPath'} = '/otrs-web/';</code>

Frontend::ImagePath

Description	Value
Description:	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>images/Standard/';</code>

Frontend::CSSPath

Description	Value
Description:	URL CSS path.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::CSSPath'} = '<OTRS_CONFIG_Frontend::WebPath>css/';</code>

Frontend::JavaScriptPath

Description	Value
Description:	URL JavaScript path.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self->{'Frontend::JavaScriptPath'} = '<OTRS_CONFIG_Frontend::WebPath>js/';</pre>

Frontend::YUIPath

Description	Value
Description:	URL YUI path.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Frontend::YUIPath'} = '<OTRS_CONFIG_Frontend::WebPath>yui/2.7.0/';</pre>

Frontend::RichText

Description	Value
Description:	Use richtext for viewing and editor for article, salutation, signature, standard_response, auto_response, notification.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Frontend::RichText'} = '1';</pre>

Frontend::RichTextWidth

Description	Value
Description:	Width for RichText editor component. Enter number (pixels) or percent value (relative).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Frontend::RichTextWidth'} = '620';</pre>

Frontend::RichTextHeight

Description	Value
Description:	Height for RichText editor component. Enter number (pixels) or percent value (relative).

Description	Value
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::RichTextHeight'} = '320';</code>

Frontend::RichText::DefaultCSS

Description	Value
Description:	Use richtext editor default CSS.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::RichText::DefaultCSS'} = 'font-family:Geneva,Helvetica,Arial,sans-serif;font-size: 12px;';</code>

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>

DefaultPreViewLines

Description	Value
Description:	Count 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>

DefaultViewLines

Description	Value
Description:	Amount 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>

AttachmentDownloadType

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

WebMaxFileUpload

Description	Value
Description:	Maximal size for file uploads via the browser (in Bytes).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'WebMaxFileUpload'} = '16777216';</code>

WebUploadCacheModule

Description	Value
Description:	Select the module to handle uploads via the web interface. "DB" stores all uploads in the database, "FS" uses the file system.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	<code>\$Self->{'WebUploadCacheModule'} = 'Kernel::System::Web::UploadCache::DB';</code>

Frontend::Output::FilterText###AAURL

Description	Value
Description:	A filter to process 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'}- >{'AAURL'} = { 'Module' => 'Kernel::Output::HTML::OutputFilterTextURL' };</code>

Frontend::Output::FilterText###OutputFilterTextAutoLink

Description	Value
Description:	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>

Frontend::Output::OutputFilterTextAutoLink###CVE

Description	Value
Description:	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>-<MATCH3>' }, '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>

Frontend::Output::OutputFilterTextAutoLink###Bugtraq

Description	Value
Description:	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
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})', </pre>

Description	Value
	<pre> '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>

Frontend::Output::OutputFilterTextAutoLink###MSBulletins

Description	Value
Description:	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>-<MATCH2>.mspx' }, 'URL2' => { 'Description' => 'Google', 'Image' => 'http://www.google.de/favicon.ico', </pre>

Description	Value
	<pre> 'Target' => '_blank', 'URL' => 'http://google.com/search? q=MS<MATCH1>--<MATCH2>' } }; </pre>

Frontend::Output::OutputFilterTextAutoLink###Setting1

Description	Value
Description:	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>

Frontend::Output::OutputFilterTextAutoLink###Setting2

Description	Value
Description:	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

Description	Value
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>

Core::WebUserAgent

WebUserAgent::Timeout

Description	Value
Description:	Timeout for http/ftp downloads (in seconds).
Group:	Framework
SubGroup:	Core::WebUserAgent
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'WebUserAgent::Timeout'} = '15'; </pre>

WebUserAgent::Proxy

Description	Value
Description:	Connections for http/ftp via a proxy.
Group:	Framework
SubGroup:	Core::WebUserAgent

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

Crypt::PGP

PGP

Description	Value
Description:	Enable 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>

PGP::Bin

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

PGP::Options

Description	Value
Description:	Options for PGP binary.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PGP::Options'} = '--homedir /opt/ otrs/.gnupg/ --batch --no-tty --yes';</code>

PGP::Key::Password

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

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>

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 successfully.', 'ERRSIG' => 'It was not possible to check the PGP signature, this may be caused by a missing public key or an unsupported algorithm.', '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 could mean that the signature is forged.'</pre>

Description	Value
	<pre>'NODATA' => 'No valid OpenPGP data found.', 'NO_PUBKEY' => 'No public key found.', 'REVKEYSIG' => 'The PGP signature was made by a revoked key, this could mean that the signature is forged.', 'SIGEXPIRED' => 'The PGP signature is expired.', 'SIG_ID' => 'Signature data.', 'TRUST_UNDEFINED' => 'This key is not certified with a trusted signature!.', 'VALIDSIG' => 'The PGP signature with the keyid is good.' };</pre>

Crypt::SMIME

SMIME

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

SMIME::Bin

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

SMIME::CertPath

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

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

SMIME::PrivatePath

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

Frontend::Admin::ModuleRegistration

Frontend::Module###Admin

Description	Value
Description:	Frontend module registration for the Admin object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'Admin'} = { 'Description' => 'Admin-Area', 'Group' => ['admin'], 'NavBar' => [{ 'AccessKey' => 'a', 'Block' => 'ItemArea', 'Description' => 'Admin-Area', 'Image' => 'admin.png', 'Link' => 'Action=Admin', 'Name' => 'Admin', 'NavBar' => 'Admin', 'Prio' => '10000', 'Type' => 'Menu' }], }</pre>

Description	Value
	<pre>'NavBarModule' => { 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin' }, 'NavBarName' => 'Admin', 'Title' => 'Admin' };</pre>

Frontend::Module###AdminInit

Description	Value
Description:	Init of a new setup.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminInit'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarName' => '', 'Title' => 'Init' };</pre>

Frontend::Module###AdminUser

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

Description	Value
	<pre> }, 'NavBarName' => 'Admin', 'Title' => 'User' }; </pre>

Frontend::Module###AdminGroup

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

Frontend::Module###AdminUserGroup

Description	Value
Description:	Frontend module registration for the AdminUserGroup object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminUserGroup'} = { 'Description' => 'Admin', 'Group' => ['admin'], </pre>

Description	Value
	<pre> 'NavBarModule' => { 'Block' => 'Block1', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Users <-> Groups', 'Prio' => '200' }, 'NavBarName' => 'Admin', 'Title' => 'Users <-> Groups' }; </pre>

Frontend::Module###AdminCustomerUser

Description	Value
Description:	Frontend module registration for the AdminCustomerUser object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminCustomerUser'} = { 'Description' => 'Edit Customer Users', 'Group' => ['admin', 'users'], 'GroupRo' => [''], 'NavBar' => [{ 'AccessKey' => 'c', 'Block' => 'ItemArea', 'Description' => 'Edit Customer Users', 'Image' => 'folder_yellow.png', 'Link' => 'Action=AdminCustomerUser&Nav=Agent', 'Name' => 'Customer', 'NavBar' => 'Customer', 'Prio' => '9000', 'Type' => 'Menu' }], 'NavBarModule' => { 'Block' => 'Block1', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Customer Users', </pre>

Description	Value
	<pre> 'Prio' => '300' }, 'NavBarName' => 'Customer', 'Title' => 'Customer User' }; </pre>

Frontend::Module###AdminCustomerCompany

Description	Value
Description:	Frontend module registration for the AdminCustomerCompany object in the admin area.
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' => 'Edit Customer Company', 'Image' => 'folder_yellow.png', 'Link' => 'Action=AdminCustomerCompany&Nav=Agent', 'Name' => 'Company', 'NavBar' => 'Ticket', 'Prio' => '9100', 'Type' => 'Menu' }], 'NavBarModule' => { 'Block' => 'Block1', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Customer Company', 'Prio' => '310' }, 'NavBarName' => 'Admin', 'Title' => 'Customer Company' }; </pre>

Frontend::Module###AdminCustomerUserGroup

Description	Value
Description:	Frontend module registration for the AdminCustomerUserGroup object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}- >{'AdminCustomerUserGroup'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block1', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Customer Users <-> Groups', 'Prio' => '400' }, 'NavBarName' => 'Admin', 'Title' => 'Customer Users <-> Groups' }; </pre>

Frontend::Module###AdminCustomerUserService

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

Description	Value
	<pre> }, 'NavBarName' => 'Admin', 'Title' => 'Customer Users <-> Services' }; </pre>

Frontend::Module###AdminRole

Description	Value
Description:	Frontend module registration for the AdminRole object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminRole'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block1', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Roles', 'Prio' => '600' }, 'NavBarName' => 'Admin', 'Title' => 'Role' }; </pre>

Frontend::Module###AdminRoleUser

Description	Value
Description:	Frontend module registration for the AdminRoleUser object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminRoleUser'} = { 'Description' => 'Admin', 'Group' => ['admin'], </pre>

Description	Value
	<pre> 'NavBarModule' => { 'Block' => 'Block1', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Roles <-> Users', 'Prio' => '700' }, 'NavBarName' => 'Admin', 'Title' => 'Roles <-> Users' }; </pre>

Frontend::Module###AdminRoleGroup

Description	Value
Description:	Frontend module registration for the AdminRoleGroup object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminRoleGroup'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block1', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Roles <-> Groups', 'Prio' => '800' }, 'NavBarName' => 'Admin', 'Title' => 'Roles <-> Groups' }; </pre>

Frontend::Module###AdminSMIME

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

Description	Value
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminSMIME'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'S/MIME', 'Prio' => '1100' }, 'NavBarName' => 'Admin', 'Title' => 'S/MIME Management' }; </pre>

Frontend::Module###AdminPGP

Description	Value
Description:	Frontend module registration for the AdminPGP object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminPGP'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'PGP', 'Prio' => '1200' }, 'NavBarName' => 'Admin', 'Title' => 'PGP Key Management' }; </pre>

Frontend::Module###AdminMailAccount

Description	Value
Description:	Frontend module registration for the AdminMailAccount object in the admin area.

Description	Value
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminMailAccount'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block4', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'PostMaster Mail Account', 'Prio' => '100' }, 'NavBarName' => 'Admin', 'Title' => 'Mail Account' }; </pre>

Frontend::Module###AdminPostMasterFilter

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

Frontend::Module###AdminEmail

Description	Value
Description:	Frontend module registration for the AdminEmail object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminEmail'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block4', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Admin Notification', 'Prio' => '400' }, 'NavBarName' => 'Admin', 'Title' => 'Admin-Email' }; </pre>

Frontend::Module###AdminSession

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

Description	Value
	<pre>'NavBarName' => 'Admin', 'Title' => 'Session Management' };</pre>

Frontend::Module###AdminPerformanceLog

Description	Value
Description:	Frontend module registration for the AdminPerformanceLog object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}- >{'AdminPerformanceLog'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block4', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Performance Log', 'Prio' => '550' }, 'NavBarName' => 'Admin', 'Title' => 'Performance Log' };</pre>

Frontend::Module###AdminLog

Description	Value
Description:	Frontend module registration for the AdminLog object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AdminLog'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block4',</pre>

Description	Value
	<pre> 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'System Log', 'Prio' => '600' }, 'NavBarName' => 'Admin', 'Title' => 'System Log' }; </pre>

Frontend::Module###AdminSelectBox

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

Frontend::Module###AdminPackageManager

Description	Value
Description:	Frontend module registration for the AdminPackageManager object in the admin area.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}- >{'AdminPackageManager'} = { </pre>

Description	Value
	<pre>'Description' => 'Software Package Manager', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block4', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Package Manager', 'Prio' => '1000' }, 'NavBarName' => 'Admin', 'Title' => 'Package Manager' };</pre>

Frontend::Agent

LoginURL

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

LogoutURL

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

PreApplicationModule###AgentInfo

Description	Value
Description:	This module is useful to load specific user options or to display news.

Description	Value
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'PreApplicationModule'}->{'AgentInfo'} = 'Kernel::Modules::AgentInfo';</code>

InfoKey

Description	Value
Description:	Key to check 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>

InfoFile

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

LostPassword

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

ShowMotd

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

NotificationSubjectLostPasswordToken

Description	Value
Description:	Subject for notification mail to agents with token about new requested password (after using this link the new password will be sent).
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'NotificationSubjectLostPasswordToken'} = 'New OTRS Password request!';</code>

NotificationBodyLostPasswordToken

Description	Value
Description:	Body for notification mail to agents with token about new requested password (after using this link the new password will be sent).
Group:	Framework
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 this link to get a new password. <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/ <OTRS_CONFIG_ScriptAlias>index.pl? Action=LostPassword&Token=<OTRS_TOKEN> </pre>

Description	Value
	Your OTRS Notification Master ';

NotificationSubjectLostPassword

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

NotificationBodyLostPassword

Description	Value
Description:	Body for notification mail to agents about new password.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'NotificationBodyLostPassword'} = 'Hi <OTRS_USERFIRSTNAME>, you or someone impersonating you has requested to change your OTRS password. New Password: <OTRS_NEWPW> <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/ <OTRS_CONFIG_ScriptAlias>index.pl Your OTRS Notification Master ';</code>

Frontend::NavBarStyle

Description	Value
Description:	With this setting you can define how the icons in the Navbar are displayed.
Group:	Framework

Description	Value
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::NavBarStyle'} = 'Modern';</code>

Frontend::NavBarStyle::ShowSelectedArea

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

Frontend::Agent::Dashboard

DashboardBackend###0000-ProductNotify

Description	Value
Description:	Parameters for the dashboard backend. "Group" are used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" means if the plugin is enabled per default or if the user needs to enable it manually. "CacheTTLLocal" means the cache time in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	<code>\$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' };</code>

DashboardBackend###0400-UserOnline

Description	Value
Description:	Parameters for the dashboard backend. "Limit" are default shown entries. "Group" are used to restricted access to the plugin (e. g. Group: admin;group1;group2;). "Default" means if the plugin is enabled per default or if the user needs to enable it manually. "CacheTTLocal" means the cache time in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
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>

DashboardBackend###0410-RSS

Description	Value
Description:	Parameters for the dashboard backend. "Limit" are default shown entries. "Group" are used to restricted access to the plugin (e. g. Group: admin;group1;group2;). "Default" means if the plugin is enabled per default or if the user needs to enable it manually. "CacheTTL" means the cache time 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' => '',</pre>

Description	Value
	<pre>'Group' => '', 'Limit' => '5', 'Module' => 'Kernel::Output::HTML::DashboardRSS', 'Title' => 'OTRS News', 'URL' => 'http://otrs.org/rss/' };</pre>

DashboardBackend###0200-Image

Description	Value
Description:	Parameters for the dashboard backend. "Group" are used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" means if the plugin is enabled per default or if the user needs to enable it manually. "CacheTTL" means the cache time in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	0
Required:	0
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>

Frontend::Agent::ModuleMetaHead

Frontend::HeaderMetaModule###1-Refresh

Description	Value
Description:	Module to generate html refresh header of html sites.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleMetaHead
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	<pre>\$Self->{'Frontend::HeaderMetaModule'}->{'1-Refresh'} = { 'Module' => 'Kernel::Output::HTML::HeaderMetaRefresh' };</pre>

Frontend::HeaderMetaModule###2-TicketSearch

Description	Value
Description:	Module to generate html OpenSearch profile for short ticket search.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleMetaHead
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::HeaderMetaModule'}->{'2-TicketSearch'} = { 'Action' => 'AgentTicketSearch', 'Module' => 'Kernel::Output::HTML::HeaderMetaTicketSearch' };</pre>

Frontend::Agent::ModuleNotify

Frontend::NotifyModule###1-CharsetCheck

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

Frontend::NotifyModule###2-UID-Check

Description	Value
Description:	Module to display a notification in the agent interface, if the system is used by the admin user (normally you shouldn't work as admin).

Description	Value
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>

Frontend::NotifyModule###3-ShowAgentOnline

Description	Value
Description:	Module to show currently logged in agents in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::NotifyModule'}->{'3- ShowAgentOnline'} = { 'IdleMinutes' => '60', 'Module' => 'Kernel::Output::HTML::NotificationAgentOnline', 'ShowEmail' => '1' };</pre>

Frontend::NotifyModule###4-ShowCustomerOnline

Description	Value
Description:	Module to show currently logged in customers in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::NotifyModule'}->{'4- ShowCustomerOnline'} = { 'IdleMinutes' => '60', 'Module' => 'Kernel::Output::HTML::NotificationCustomerOnline', 'ShowEmail' => '1' };</pre>

Frontend::Agent::ModuleRegistration

Frontend::Module###Logout

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

Frontend::Module###AgentDashboard

Description	Value
Description:	Frontend module registration for the AgentDashboard object in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentDashboard'} = { 'Description' => 'Agent Dashboard', 'NavBar' => [{ 'AccessKey' => 'd', 'Block' => 'ItemArea', </pre>

Description	Value
	<pre> 'Description' => 'Agent Dashboard', 'Image' => 'dashboard.png', 'Link' => 'Action=AgentDashboard', 'Name' => 'Dashboard', 'NavBar' => 'Dashboard', 'Prio' => '50', 'Type' => 'Menu' }], 'NavBarName' => 'Dashboard', 'Title' => '' }; </pre>

Frontend::Module###AgentPreferences

Description	Value
Description:	Frontend module registration for the AgentPreference object in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentPreferences'} = { 'Description' => 'Agent Preferences', 'NavBar' => [{ 'AccessKey' => 'p', 'Block' => 'ItemArea', 'Description' => 'Agent Preferences', 'Image' => 'prefer.png', 'Link' => 'Action=AgentPreferences', 'Name' => 'Preferences', 'NavBar' => 'Preferences', 'Prio' => '9900', 'Type' => 'Menu' }], 'NavBarName' => 'Preferences', 'Title' => '' }; </pre>

Frontend::Module###PictureUpload

Description	Value
Description:	Frontend module registration for the PictureUpload object in the agent interface.

Description	Value
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>

Frontend::Module###AgentSpelling

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

Frontend::Module###SpellingInline

Description	Value
Description:	Frontend module registration for the SpellingInline object in 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>

Frontend::Module###AgentBook

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

Frontend::Module###AgentLookup

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

Frontend::Module###AgentLinkObject

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

Description	Value
	<pre>'NavBarName' => '', 'Title' => 'Link Object' };</pre>

Frontend::Module###AgentInfo

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

Frontend::Module###AgentCalendarSmall

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

CustomerFrontend::Module###SpellingInline

Description	Value
Description:	Frontend module registration for the SpellingInline object in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration

Description	Value
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}->{'SpellingInline'} = { 'Description' => 'Spell checker', 'NavBarName' => '', 'Title' => 'Spell Checker' };</pre>

Frontend::Module###AgentStats

Description	Value
Description:	FrontendModuleRegistration for Stats to define the registration settings in the program and to set rights and positions of icons in the navigation bar.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentStats'} = { 'Description' => 'Stats', 'Group' => ['stats'], 'GroupRo' => ['stats'], 'NavBar' => [{ 'AccessKey' => '', 'Block' => 'ItemArea', 'Description' => 'Stats', 'Image' => 'stats.png', 'Link' => 'Action=AgentStats&Subaction=Overview', 'Name' => 'Stats', 'NavBar' => 'Stats', 'Prio' => '8500', 'Type' => 'Menu' }, { 'AccessKey' => '', 'Block' => '', 'Description' => 'Overview', 'GroupRo' => ['stats'], }], }</pre>

Description	Value
	<pre> 'Image' => 'overview.png', 'Link' => 'Action=AgentStats&Subaction=Overview', 'Name' => 'Overview', 'NavBar' => 'Stats', 'Prio' => '100', 'Type' => '' }, { 'AccessKey' => '', 'Block' => '', 'Description' => 'New', 'Group' => ['stats'], 'Image' => 'new.png', 'Link' => 'Action=AgentStats&Subaction=Add', 'Name' => 'New', 'NavBar' => 'Stats', 'Prio' => '200', 'Type' => '' }, { 'AccessKey' => '', 'Block' => '', 'Description' => 'Import', 'Group' => ['stats'], 'Image' => 'import.png', 'Link' => 'Action=AgentStats&Subaction=Import', 'Name' => 'Import', 'NavBar' => 'Stats', 'Prio' => '300', 'Type' => '' }], 'NavBarName' => 'Stats', 'Title' => 'Stats' }; </pre>

Frontend::Agent::NavBarModule

Frontend::NavBarModule###6-CustomerCompany

Description	Value
Description:	Module registration for the AdminCustomerCompany object in the agent interface (disable Company link if no Company Feature is used).
Group:	Framework

Description	Value
SubGroup:	Frontend::Agent::NavBarModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::NavBarModule'}->{'6-CustomerCompany'} = { 'Module' => 'Kernel::Output::HTML::NavBarCustomerCompany' };</pre>

Frontend::Agent::Preferences

PreferencesTableValue

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

PreferencesTableUserID

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

PreferencesView

Description	Value
Description:	Display order of the different items in the preference view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self->{'PreferencesView'} = ['Frontend', 'Mail Management', 'Other Options'];</pre>

PreferencesGroups###Password

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

PreferencesGroups###SpellDict

Description	Value
Description:	Parameters for the SpellDict object in the preference view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'SpellDict'} = { 'Activ' => '1', 'Colum' => 'Other Options', 'Data' => { 'deutsch' => 'Deutsch', 'english' => 'English' } };</pre>

Description	Value
	<pre> }, 'DataSelected' => 'english', 'Desc' => 'Select your default spelling dictionary.', 'Label' => 'Spelling Dictionary', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserSpellDict', 'Prio' => '5000' }; </pre>

PreferencesGroups###Comment

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

PreferencesGroups###FreeText

Description	Value
Description:	Parameters for the Freetext object in the preference view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'FreeText'} = { 'Activ' => '1', 'Block' => 'Input', 'Colum' => 'Other Options', 'Data' => '\$Env{"UserFreeText"}', </pre>

Description	Value
	<pre>'Desc' => 'Example for free text', 'Label' => 'Comment', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserFreeText', 'Prio' => '7000' };</pre>

PreferencesGroups###Language

Description	Value
Description:	Parameters for the Language object in the preference view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'Language'} = { 'Activ' => '1', 'Colum' => 'Frontend', 'Desc' => 'Select your frontend language.', 'Label' => 'Language', 'Module' => 'Kernel::Output::HTML::PreferencesLanguage', 'PrefKey' => 'UserLanguage', 'Prio' => '1000' };</pre>

PreferencesGroups###Theme

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

PreferencesGroups###TimeZone

Description	Value
Description:	Parameters for the Time-Zone object in the preference view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'TimeZone'} = { 'Activ' => '1', 'Colum' => 'Frontend', 'Desc' => 'Select your Time-Zone.', 'Label' => 'Time-Zone', 'Module' => 'Kernel::Output::HTML::PreferencesTimeZone', 'PrefKey' => 'UserTimeZone', 'Prio' => '3000' };</pre>

PreferencesGroups###OutOfOffice

Description	Value
Description:	Parameters for the Out Of Office object in the preference view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'OutOfOffice'} = { 'Activ' => '1', 'Block' => 'OutOfOffice', 'Colum' => 'Other Options', 'Desc' => 'Select your out of office time.', 'Label' => 'Out Of Office', 'Module' => 'Kernel::Output::HTML::PreferencesOutOfOffice', 'PrefKey' => 'UserOutOfOffice', 'Prio' => '4000' };</pre>

Frontend::Agent::Stats

Stats::SearchPageShown

Description	Value
Description:	Default maximum number of search results shown on the Overview page.

Description	Value
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Stats::SearchPageShown'} = '20';</code>

Stats::DefaultSelectedDynamicObject

Description	Value
Description:	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>

Stats::DefaultSelectedPermissions

Description	Value
Description:	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>

Stats::DefaultSelectedFormat

Description	Value
Description:	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

Description	Value
Required:	1
Config-Setting:	<pre>\$Self->{'Stats::DefaultSelectedFormat'} = ['Print', 'CSV'];</pre>

Stats::SearchLimit

Description	Value
Description:	Search Limit.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Stats::SearchLimit'} = '500';</pre>

Stats::Format

Description	Value
Description:	Possible output formats.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	<pre>\$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' };</pre>

Stats::GraphSize

Description	Value
Description:	Size of stat graph.
Group:	Framework

Description	Value
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Stats::GraphSize'} = { '1200x800' => '1200x800', '1600x1200' => '1600x1200', '800x600' => '800x600' };</pre>

Stats::TimeType

Description	Value
Description:	Selection of 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>

Stats::ExchangeAxis

Description	Value
Description:	If you activate this option an agent can exchange the axis of a stat if he generates one.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Stats::ExchangeAxis'} = '0';</pre>

Stats::UseAgentElementInStats

Description	Value
Description:	If you activate this option you can generate individual-related stats.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Stats::UseAgentElementInStats'} = '0';</pre>

Stats::CustomerIDAsMultiSelect

Description	Value
Description:	If you activate this option all CustomerIDs will be shown in a multi-select field (Not useful if you have a lot of CustomerIDs).
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Stats::CustomerIDAsMultiSelect'} = '1';</code>

Frontend::Customer

CustomerPanelUserID

Description	Value
Description:	UserID for customer panel.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelUserID'} = '1';</code>

CustomerGroupSupport

Description	Value
Description:	Activate support for customer groups?
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerGroupSupport'} = '0';</code>

CustomerGroupAlwaysGroups

Description	Value
Description:	If CustomerGroupSupport is enabled and you don't want to manage every user for these groups, then put the groups for every customer user in there.
Group:	Framework

Description	Value
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerGroupAlwaysGroups'} = ['users', 'info'];</pre>

CustomerPanelLoginURL

Description	Value
Description:	Alternate login URL for the customer panel..
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerPanelLoginURL'} = 'http:// host.example.com/cgi-bin/login.pl';</pre>

CustomerPanelLogoutURL

Description	Value
Description:	Alternate logout URL for the customer panel.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerPanelLogoutURL'} = 'http:// host.example.com/cgi-bin/login.pl';</pre>

Frontend::CustomerUser::Item###1-GoogleMaps

Description	Value
Description:	A customer item (icon) which generates an google maps icon at the end of an 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;', 'Image' => 'maps.png', 'Module' => 'Kernel::Output::HTML::CustomerUserGeneric', 'Required' => 'UserStreet;UserCity;', 'Target' => '_blank', 'Text' => 'Location', 'URL' => 'http://maps.google.com/maps?z=7&q=' }; </pre>

Frontend::CustomerUser::Item###2-Google

Description	Value
Description:	A customer item (icon) which generates an google icon at the end of an customer info block.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::CustomerUser::Item'}->{'2-Google'} = { 'Attributes' => 'UserFirstname;UserLastname;', 'Image' => 'maps.png', 'Module' => 'Kernel::Output::HTML::CustomerUserGeneric', 'Required' => 'UserFirstname;UserLastname;', 'Target' => '_blank', 'Text' => 'Google', 'URL' => 'http://google.com/search?q=' }; </pre>

Frontend::CustomerUser::Item###2-LinkedIn

Description	Value
Description:	A customer item (icon) which generates an LinkedIn icon at the end of an customer info block.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	<pre>\$Self->{'Frontend::CustomerUser::Item'}->{'2-LinkedIn'} = { 'Attributes' => 'UserFirstname;UserLastname;', 'Image' => 'in.png', 'Module' => 'Kernel::Output::HTML::CustomerUserGeneric', 'Required' => 'UserFirstname;UserLastname;', 'Target' => '_blank', 'Text' => 'LinkedIn', 'URL' => 'http://www.linkedin.com/commonSearch? type=people&keywords=' };</pre>

Frontend::CustomerUser::Item###3-XING

Description	Value
Description:	A customer item (icon) which generates an XING icon at the end of an customer info block.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::CustomerUser::Item'}->{'3-XING'} = { 'Attributes' => 'UserFirstname;UserLastname;', 'Image' => 'xing.png', 'Module' => 'Kernel::Output::HTML::CustomerUserGeneric', 'Required' => 'UserFirstname;UserLastname;', 'Target' => '_blank', 'Text' => 'XING', 'URL' => 'https://www.xing.com/app/search? op=search&keywords=' };</pre>

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
Config-Setting:	

Description	Value
	<pre>\$Self->{'CustomerPanelPreApplicationModule'}- >{'CustomerAccept'} = 'Kernel::Modules::CustomerAccept';</pre>

CustomerPanel::InfoKey

Description	Value
Description:	Key to check with CustomerAccept. If this user preferences key is true, then the message is accepted by the system.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerPanel::InfoKey'} = 'CustomerAccept1';</pre>

CustomerPanel::InfoFile

Description	Value
Description:	Shown InfoFile, that is located under Kernel/Output/HTML/Standard/CustomerAccept.dtl.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerPanel::InfoFile'} = 'CustomerAccept';</pre>

CustomerPanelLostPassword

Description	Value
Description:	Activate lost password feature for customers?
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerPanelLostPassword'} = '1';</pre>

CustomerPanelCreateAccount

Description	Value
Description:	Can Customers create their own account?

Description	Value
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelCreateAccount'} = '1';</code>

CustomerPanelSubjectLostPasswordToken

Description	Value
Description:	Subject for notification mail 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->{'CustomerPanelSubjectLostPasswordToken'} = 'New OTRS Password request!';</code>

CustomerPanelBodyLostPasswordToken

Description	Value
Description:	Body for notification mail 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:	<pre> \$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 to get a new password. <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/ <OTRS_CONFIG_ScriptAlias>customer.pl? Action=CustomerLostPassword&Token=<OTRS_TOKEN> Your OTRS Notification Master </pre>

Description	Value
	';

CustomerPanelSubjectLostPassword

Description	Value
Description:	Subject of notification mail to customers about new password.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelSubjectLostPassword'} = 'New OTRS Password!';</code>

CustomerPanelBodyLostPassword

Description	Value
Description:	Body of notification mail for customers about new password.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelBodyLostPassword'} = 'Hi <OTRS_USERFIRSTNAME>, you or someone impersonating you has requested to change your OTRS password. New Password: <OTRS_NEWPW> <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/<OTRS_CONFIG_ScriptAlias>customer.pl Your OTRS Notification Master ';</code>

CustomerPanelSubjectNewAccount

Description	Value
Description:	Subject for notification email to customers about new account.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1

Description	Value
Required:	1
Config-Setting:	<code>\$Self->{'CustomerPanelSubjectNewAccount'} = 'New OTRS Account!';</code>

CustomerPanelBodyNewAccount

Description	Value
Description:	Body of notification email to customer 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 (<OTRS_USERFIRSTNAME> <OTRS_USERLASTNAME>). Login: <OTRS_USERLOGIN> Password: <OTRS_USERPASSWORD> <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/ <OTRS_CONFIG_ScriptAlias>customer.pl Your OTRS Notification Master '; </pre>

Frontend::CustomerUser::Item###9-OpenTickets

Description	Value
Description:	A customer item (icon) which shows the open Tickets of this customer as customer info block.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::CustomerUser::Item'}->{'9- OpenTickets'} = { 'Action' => 'AgentTicketSearch', 'Attributes' => 'StateType=Open;', 'ImageNoOpenTicket' => 'greenled-small.png', 'ImageOpenTicket' => 'redled-small.png', </pre>

Description	Value
	<pre>'Module' => 'Kernel::Output::HTML::CustomerUserGenericTicket', 'Subaction' => 'Search', 'Target' => '_blank', 'Text' => 'Open Tickets' };</pre>

Frontend::Customer::Auth

Customer::AuthModule

Description	Value
Description:	Module to authenticate customers.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::DB';</pre>

Customer::AuthModule::DB::CryptType

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, you have to specify the crypt type of passwords.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::DB::CryptType'} = 'md5';</pre>

Customer::AuthModule::DB::Table

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, you have to specify the name of the table where your customer data are stored.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::Table'} = 'customer_user';</code>

Customer::AuthModule::DB::CustomerKey

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, you have to specify the name of the column for the CustomerKey in your customer table.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::CustomerKey'} = 'login';</code>

Customer::AuthModule::DB::CustomerPassword

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, you have to specify the column name for the CustomerPassword in your customer table.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::CustomerPassword'} = 'pw';</code>

Customer::AuthModule::DB::DSN

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, you hav to specify the DSN for the connection to the customer table.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::DB::DSN'} = 'DBI:mysql:database=customerdb;host=customerdbhost';</code>

Customer::AuthModule::DB::User

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, you can specify a username to connect to your customer table.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::DB::User'} = 'some_user';</pre>

Customer::AuthModule::DB::Password

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, you can specify a password to connect to your customer table.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::DB::Password'} = 'some_password';</pre>

Customer::AuthModule::DB::Type

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, you can specify a database driver (normally autodetection is used).
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::DB::Type'} = 'mysql';</pre>

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).

Description	Value
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self- >{ 'Customer::AuthModule::HTTPBasicAuth::Replace' } = 'example_domain\\\\';</pre>

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:	<pre>\$Self- >{ 'Customer::AuthModule::HTTPBasicAuth::ReplaceRegExp' } = '^(.+?)@.+?\$';</pre>

Customer::AuthModule::LDAP::Host

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, you can specify your LDAP host.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{ 'Customer::AuthModule::LDAP::Host' } = 'ldap.example.com';</pre>

Customer::AuthModule::LDAP::BaseDN

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, specify your BaseDN.
Group:	Framework

Description	Value
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';</code>

Customer::AuthModule::LDAP::UID

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, specify the UID.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::UID'} = 'uid';</code>

Customer::AuthModule::LDAP::GroupDN

Description	Value
Description:	If "LDAP" was selected for Customer::Authmodule, you can check if the user is allowed to auth because he is in a posixGroup, e.g. user needs to be in a group xyz to use otrs. Specify the group, who may access the system.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Customer::AuthModule::LDAP::GroupDN'} = 'cn=otrsallow,ou=posixGroups,dc=example,dc=com';</code>

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

Description	Value
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::LDAP::AccessAttr'} = 'memberUid';</pre>

Customer::AuthModule::LDAP::UserAttr

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, you can specify user attributes. For LDAP posixGroups objectclass UID, for non ldap posixGroups objectclass full user DN.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::LDAP::UserAttr'} = 'UID';</pre>

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:	<pre>\$Self->{'Customer::AuthModule::LDAP::SearchUserDN'} = 'cn=binduser,ou=users,dc=example,dc=com';</pre>

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

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::LDAP::SearchUserPw'} = 'some_password';</pre>

Customer::AuthModule::LDAP::AlwaysFilter

Description	Value
Description:	If "LDAP" was selected, you can add a filter to each LDAP query, e.g. (mail=*), (objectclass=user) or (!objectclass=computer).
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::LDAP::AlwaysFilter'} = '(!objectclass=computer)';</pre>

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:	<pre>\$Self->{'Customer::AuthModule::LDAP::UserSuffix'} = '@domain.com';</pre>

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

Description	Value
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::LDAP::Params'} = { 'async' => '0', 'port' => '389', 'timeout' => '120', 'version' => '3' };</pre>

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 server can't connected (e. g. network problems).
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::LDAP::Die'} = '1';</pre>

Customer::AuthModule::Radius::Host

Description	Value
Description:	If "Radius" was selected for Customer::AuthModule, specify your radius host.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::Radius::Host'} = 'radiushost';</pre>

Customer::AuthModule::Radius::Password

Description	Value
Description:	If "Radius" was selected for Customer::AuthModule, specify the password to authenticate to your radius host.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::Radius::Password'} = 'radiussecret';</pre>

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 server can't connected (e. g. network problems).
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Customer::AuthModule::Radius::Die'} = '1';</pre>

Frontend::Customer::ModuleMetaHead

CustomerFrontend::HeaderMetaModule###1-Refresh

Description	Value
Description:	Module to generate html refresh header of html sites.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleMetaHead
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerFrontend::HeaderMetaModule'}- >{'1-Refresh'} = { 'Module' => 'Kernel::Output::HTML::HeaderMetaRefresh' };</pre>

CustomerFrontend::HeaderMetaModule###2-TicketSearch

Description	Value
Description:	Module to generate html OpenSearch profile for short ticket search.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleMetaHead
Valid:	1

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::HeaderMetaModule'}->{'2-TicketSearch'} = { 'Action' => 'CustomerTicketSearch', 'Module' => 'Kernel::Output::HTML::HeaderMetaTicketSearch' };</pre>

Frontend::Customer::ModuleNotify

CustomerFrontend::NotifyModule###1-ShowAgentOnline

Description	Value
Description:	Module that shows the currently logged in agents in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::NotifyModule'}->{'1-ShowAgentOnline'} = { 'IdleMinutes' => '60', 'Module' => 'Kernel::Output::HTML::NotificationAgentOnline', 'ShowEmail' => '1' };</pre>

CustomerFrontend::NotifyModule###1-ShowCustomerOnline

Description	Value
Description:	Module that shows the currently logged in customers in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::NotifyModule'}->{'1-ShowCustomerOnline'} = { 'Module' => 'Kernel::Output::HTML::NotificationCustomerOnline', 'ShowEmail' => '1' };</pre>

Frontend::Customer::ModuleRegistration

CustomerFrontend::Module###Logout

Description	Value
Description:	Frontend module registration for the Logout object in the customer frontend.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerFrontend::Module'}->{'Logout'} = { 'Description' => 'Logout of customer panel', 'NavBar' => [{ 'AccessKey' => 'l', 'Block' => '', 'Description' => 'Logout', 'Image' => 'exit.png', 'Link' => 'Action=Logout', 'Name' => 'Logout', 'NavBar' => '', 'Prio' => '10', 'Type' => '' }], 'NavBarName' => '', 'Title' => '' }; </pre>

CustomerFrontend::Module###CustomerPreferences

Description	Value
Description:	Frontend module registration for the CustomerPreferences object in the customer frontend.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerFrontend::Module'}- >{'CustomerPreferences'} = { 'Description' => 'Customer preferences', 'NavBar' => [{ 'AccessKey' => 'p', </pre>

Description	Value
	<pre> 'Block' => '', 'Description' => 'Preferences', 'Image' => 'prefer.png', 'Link' => 'Action=CustomerPreferences', 'Name' => 'Preferences', 'NavBar' => '', 'Prio' => '1000', 'Type' => '' }], 'NavBarName' => '', 'Title' => 'Preferences' }; </pre>

CustomerFrontend::Module###CustomerCalendarSmall

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

CustomerFrontend::Module###CustomerAccept

Description	Value
Description:	Frontend module registration for the CustomerAccept object in the customer frontend.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerFrontend::Module'}- >{'CustomerAccept'} = { 'Description' => 'To accept login infos', 'NavBarName' => '', 'Title' => 'Info' }; </pre>

Description	Value
	<code>};</code>

CustomerFrontend::Module###PictureUpload

Description	Value
Description:	Customer frontend module registration for the PictureUpload object in the agent 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>

Frontend::Customer::Preferences

PreferencesTable

Description	Value
Description:	The name of the table, where the customer preferences are stored.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'PreferencesTable'} = 'user_preferences';</pre>

PreferencesTableKey

Description	Value
Description:	Key for preferences table.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'PreferencesTableKey'} = 'preferences_key';</pre>

CustomerPreferences

Description	Value
Description:	Parameters for the customer preference table.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'CustomerPreferences'} = { 'Module' => 'Kernel::System::CustomerUser::Preferences::DB', 'Params' => { 'Table' => 'customer_preferences', 'TableKey' => 'preferences_key', 'TableUserID' => 'user_id', 'TableValue' => 'preferences_value' } }; </pre>

CustomerPreferencesView

Description	Value
Description:	Order of the different items in the customer preferences view.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'CustomerPreferencesView'} = ['Frontend', 'Other Options']; </pre>

CustomerPreferencesGroups###Password

Description	Value
Description:	All parameters for the password area in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerPreferencesGroups'}->{'Password'} = { 'Activ' => '1', </pre>

Description	Value
	<pre> 'Area' => 'Customer', 'Colum' => 'Other Options', 'Label' => 'Change Password', 'Module' => 'Kernel::Output::HTML::PreferencesPassword', 'PasswordMin2Characters' => '0', 'PasswordMin2Lower2UpperCharacters' => '0', 'PasswordMinSize' => '0', 'PasswordNeedDigit' => '0', 'PasswordRegExp' => '', 'Prio' => '1000' }; </pre>

CustomerPreferencesGroups###Language

Description	Value
Description:	All parameters for the language area in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerPreferencesGroups'}->{'Language'} = { 'Activ' => '1', 'Colum' => 'Frontend', 'Desc' => 'Select your frontend language.', 'Label' => 'Language', 'Module' => 'Kernel::Output::HTML::PreferencesLanguage', 'PrefKey' => 'UserLanguage', 'Prio' => '2000' }; </pre>

CustomerPreferencesGroups###Theme

Description	Value
Description:	All parameters for the theme area in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerPreferencesGroups'}->{'Theme'} = { 'Activ' => '0', 'Colum' => 'Frontend', 'Desc' => 'Select your frontend Theme.', </pre>

Description	Value
	<pre>'Label' => 'Theme', 'Module' => 'Kernel::Output::HTML::PreferencesTheme', 'PrefKey' => 'UserTheme', 'Prio' => '1000' };</pre>

CustomerPreferencesGroups###PGP

Description	Value
Description:	All parameters for the PGP area in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerPreferencesGroups'}->{'PGP'} = { 'Activ' => '1', 'Colum' => 'Other Options', 'Desc' => 'PGP Key Upload', 'Label' => 'PGP Key', 'Module' => 'Kernel::Output::HTML::PreferencesPGP', 'PrefKey' => 'UserPGPKey', 'Prio' => '10000' };</pre>

CustomerPreferencesGroups###SMIME

Description	Value
Description:	All parameters for the S/MIME area in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerPreferencesGroups'}->{'SMIME'} = { 'Activ' => '1', 'Colum' => 'Other Options', 'Desc' => 'S/MIME Certificate Upload', 'Label' => 'S/MIME Certificate', 'Module' => 'Kernel::Output::HTML::PreferencesSMIME', 'PrefKey' => 'UserSMIMEKey', 'Prio' => '11000' };</pre>

Frontend::Public

PublicFrontend::CommonParam###Action

Description	Value
Description:	Default value for the Action parameter for the public frontend. The Action parameter is used in the scripts of the system.
Group:	Framework
SubGroup:	Frontend::Public
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PublicFrontend::CommonParam'}->{'Action'} = 'PublicDefault';</code>

Frontend::Public::ModuleRegistration

PublicFrontend::Module###PublicDefault

Description	Value
Description:	Frontend module registration for the PublicDefault object in the public interface.
Group:	Framework
SubGroup:	Frontend::Public::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'PublicFrontend::Module'}->{'PublicDefault'} = { 'Description' => 'PublicDefault', 'NavBarName' => '', 'Title' => 'PublicDefault' };</code>

PublicFrontend::Module###PublicRepository

Description	Value
Description:	Frontend module registration for the PublicRepository object in the public interface.
Group:	Framework
SubGroup:	Frontend::Public::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	<pre>\$Self->{'PublicFrontend::Module'}- >{'PublicRepository'} = { 'Description' => 'PublicRepository', 'NavBarName' => '', 'Title' => 'PublicRepository' };</pre>

Ticket

Core::ArticleFreeText

ArticleFreeKey1

Description	Value
Description:	Define the free key field 1 for articles. With this setting you can define a new article property.
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'ArticleFreeKey1'} = { 'Work' => 'Work' };</pre>

ArticleFreeKey1::DefaultSelection

Description	Value
Description:	The default selection of article FreeTextField 1 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'ArticleFreeKey1::DefaultSelection'} = '';</pre>

ArticleFreeText1

Description	Value
Description:	Define the FreeTextField 1 for article. It's a new article property.
Group:	Ticket
SubGroup:	Core::ArticleFreeText

Description	Value
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'ArticleFreeText1'} = { '' => '-', 'Bugfix' => 'Bugfix', 'Consulting' => 'Consulting', 'Research' => 'Research' };</pre>

ArticleFreeText1::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 1 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'ArticleFreeText1::DefaultSelection'} = 'Research';</pre>

ArticleFreeKey2

Description	Value
Description:	Define the free key field 2 for articles. It's a new article property.
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'ArticleFreeKey2'} = { '' => '-', 'Work' => 'Work' };</pre>

ArticleFreeKey2::DefaultSelection

Description	Value
Description:	The default selection of free key field 2 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0

Description	Value
Required:	0
Config-Setting:	<code>\$Self->{'ArticleFreeKey2::DefaultSelection'} = '';</code>

ArticleFreeText2

Description	Value
Description:	Define the FreeTextField 2 for articles. It's a new article property.
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'ArticleFreeText2'} = { ' => '-', 'Bugfix' => 'Bugfix', 'Consulting' => 'Consulting', 'Research' => 'Research' };</code>

ArticleFreeText2::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 2 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'ArticleFreeText2::DefaultSelection'} = 'Research';</code>

ArticleFreeKey3

Description	Value
Description:	Define the free key field 3 for articles. It's a new article property.
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'ArticleFreeKey3'} = {</code>

Description	Value
	<pre>' ' => '-', 'Work' => 'Work' };</pre>

ArticleFreeKey3::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 3 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'ArticleFreeKey3::DefaultSelection'} = '';</pre>

ArticleFreeText3

Description	Value
Description:	Define the FreeTextField 3 for article. It's a new article ticket property.
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'ArticleFreeText3'} = { ' ' => '-', 'Bugfix' => 'Bugfix', 'Consulting' => 'Consulting', 'Research' => 'Research' };</pre>

ArticleFreeText3::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 3 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::ArticleFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'ArticleFreeText3::DefaultSelection'} = 'Research';</pre>

Core::FulltextSearch

Ticket::SearchIndexModule

Description	Value
Description:	Select your TicketSearchIndexModule to increase your full text search of article (From, To, Cc, Subject and Body search). Runtime will do fulltext searches on live data on runtime (fine till about 50.000 tickets). StaticDB will strip all articles and will build and index (all uniq words of an article) after article creation (this will increase fulltext searches about 50%). To create 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::ArticleSearchIndex::RuntimeDB';</pre>

Ticket::SearchIndex::Attribute

Description	Value
Description:	Attributes to configure fulltext index (note: after changeing one, "bin/otrs.RebuildFulltextIndex.pl" need to be executed to generate 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>

Ticket::EventModulePost###98-ArticleSearchIndex

Description	Value
Description:	Ticket event module to build article index on the fly (directly after article creation).
Group:	Ticket
SubGroup:	Core::FulltextSearch
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}->{'98- ArticleSearchIndex'} = { 'Event' => ' (ArticleCreate ArticleUpdate) ', 'Module' => 'Kernel::System::Ticket::Event::ArticleSearchIndex' };</pre>

Core::PostMaster

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:	<pre>\$Self->{'PostmasterMaxEmails'} = '40';</pre>

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
Required:	1
Config-Setting:	<pre>\$Self->{'PostMasterMaxEmailSize'} = '16384';</pre>

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:	

Description	Value
	<code>\$Self->{'PostMasterReconnectMessage'} = '20';</code>

LoopProtectionModule

Description	Value
Description:	Default loop protection modul.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LoopProtectionModule'} = 'Kernel::System::PostMaster::LoopProtection::DB';</code>

LoopProtectionLog

Description	Value
Description:	If "FS" was selected for LoopProtectionModule, a logfile has to be specified.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'LoopProtectionLog'} = '<OTRS_CONFIG_Home>/var/log/LoopProtection';</code>

PostmasterAutoHTML2Text

Description	Value
Description:	Would you like to convert HTML mails into text messages?
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterAutoHTML2Text'} = '1';</code>

PostmasterFollowUpSearchInReferences

Description	Value
Description:	Would you like to execute followup checks on In-Reply-To or References headers for mails, that don't have a ticket number in the subject?

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

PostmasterFollowUpSearchInBody

Description	Value
Description:	Would you like to execute followup checks in mail body, 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>

PostmasterFollowUpSearchInAttachment

Description	Value
Description:	Would you like to execute followup checks in mail attachments, that don't have a ticket number in the subject?
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterFollowUpSearchInAttachment'} = '0';</code>

PostmasterFollowUpSearchInRaw

Description	Value
Description:	Would you like to execute followup checks in mail plain/raw, that don't have a ticket number in the subject?
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'PostmasterFollowUpSearchInRaw'} = '0';</code>

PostmasterUserID

Description	Value
Description:	Specify the postmaster DB UserID.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterUserID'} = '1';</code>

PostmasterDefaultQueue

Description	Value
Description:	The postmaster default queue.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterDefaultQueue'} = 'Raw';</code>

PostmasterDefaultPriority

Description	Value
Description:	The default priority of new tickets.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterDefaultPriority'} = '3 normal';</code>

PostmasterDefaultState

Description	Value
Description:	Default state of new tickets.
Group:	Ticket

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

PostmasterFollowUpState

Description	Value
Description:	The state if a ticket got a follow-up.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterFollowUpState'} = 'open';</code>

PostmasterFollowUpStateClosed

Description	Value
Description:	The state if a ticket got a follow-up and the ticket was already closed.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'PostmasterFollowUpStateClosed'} = 'open';</code>

PostmasterFollowUpOnUnlockAgentNotifyOnlyToOwner

Description	Value
Description:	Send agent follow-up notification just to the owner, if a ticket is unlocked? The default is to send a notification to all agents, if a ticket is unlocked.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'PostmasterFollowUpOnUnlockAgentNotifyOnlyToOwner'} = '0';</code>

PostmasterX-Header

Description	Value
Description:	All X-headers that should be scanned.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
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', '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',</pre>

Description	Value
	'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',

Description	Value
	<pre> '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', 'X-OTRS-FollowUp-TicketKey4', 'X-OTRS-FollowUp-TicketKey5', 'X-OTRS-FollowUp-TicketKey6', 'X-OTRS-FollowUp-TicketKey7', 'X-OTRS-FollowUp-TicketKey8', 'X-OTRS-FollowUp-TicketKey9', 'X-OTRS-FollowUp-TicketKey10', 'X-OTRS-FollowUp-TicketKey11', 'X-OTRS-FollowUp-TicketKey12', 'X-OTRS-FollowUp-TicketKey13', 'X-OTRS-FollowUp-TicketKey14', 'X-OTRS-FollowUp-TicketKey15', 'X-OTRS-FollowUp-TicketKey16', 'X-OTRS-FollowUp-TicketValue1', 'X-OTRS-FollowUp-TicketValue2', 'X-OTRS-FollowUp-TicketValue3', 'X-OTRS-FollowUp-TicketValue4', 'X-OTRS-FollowUp-TicketValue5', 'X-OTRS-FollowUp-TicketValue6', 'X-OTRS-FollowUp-TicketValue7', 'X-OTRS-FollowUp-TicketValue8', 'X-OTRS-FollowUp-TicketValue9', 'X-OTRS-FollowUp-TicketValue10', 'X-OTRS-FollowUp-TicketValue11', 'X-OTRS-FollowUp-TicketValue12', 'X-OTRS-FollowUp-TicketValue13', 'X-OTRS-FollowUp-TicketValue14', 'X-OTRS-FollowUp-TicketValue15', 'X-OTRS-FollowUp-TicketValue16', 'X-OTRS-FollowUp-TicketTime1', 'X-OTRS-FollowUp-TicketTime2', 'X-OTRS-FollowUp-TicketTime3', 'X-OTRS-FollowUp-TicketTime4', 'X-OTRS-FollowUp-TicketTime5', 'X-OTRS-FollowUp-TicketTime6']; </pre>

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>

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>

PostMaster::PreFilterModule###3-NewTicketReject

Description	Value
Description:	Block all incoming emails without 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>

PostMaster::PreFilterModule::NewTicketReject::Subject

Description	Value
Description:	Subject for NewTicketReject-Filter email.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'PostMaster::PreFilterModule::NewTicketReject::Subject'} = 'Email Rejected';</pre>

PostMaster::PreFilterModule::NewTicketReject::Body

Description	Value
Description:	Body for NewTicketReject-Filter email.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre> \$Self->{'PostMaster::PreFilterModule::NewTicketReject::Body'} = ' Dear Customer, unfortunately you have no 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>

PostMaster::PreFilterModule###4-CMD

Description	Value
Description:	CMD example setup. Ignore 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>

PostMaster::PreFilterModule###5-SpamAssassin

Description	Value
Description:	Spam Assassin example setup. Ignore emails that are marked with SpamAssassin.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	<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>

PostMaster::PreFilterModule###6-SpamAssassin

Description	Value
Description:	Spam Assassin example setup. Move marked mails to spam queue.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
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>

PostMaster::PreFilterModule###000-MatchDBSource

Description	Value
Description:	Module to use database filter storage (use it at first).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'PostMaster::PreFilterModule'}->{'000-MatchDBSource'} = { 'Module' => 'Kernel::System::PostMaster::Filter::MatchDBSource' };</pre>

PostMaster::PostFilterModule###000-FollowUpArticleTypeCheck

Description	Value
Description:	Module to check if arrived email should be marked als email-internal (because of original forwarded internal email it college). ArticleType and SenderType are to 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>

SendNoAutoResponseRegExp

Description	Value
Description:	If this regex matches, no message will be send by the autoresponder.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'SendNoAutoResponseRegExp'} = '(MAILER-DAEMON postmaster abuse)@.+?\ .+?';</pre>

Core::Ticket

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:	

Description	Value
	<code>\$Self->{'Ticket::Hook'} = 'Ticket#';</code>

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>

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>

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>

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.

Description	Value
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::CustomQueue'} = 'My Queues';</code>

Ticket::ChangeOwnerToEveryone

Description	Value
Description:	If you would like to make it possible to change 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>

Ticket::NewMessageMode

Description	Value
Description:	Mode how the new message count in the agent navigation bar is used.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::NewMessageMode'} = 'ArticleLastSender';</code>

Ticket::Responsible

Description	Value
Description:	Enable ticket responsible feature.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	<code>\$Self->{'Ticket::Responsible'} = '0';</code>

Ticket::ResponsibleAutoSet

Description	Value
Description:	If ticket responsible feature is enabled, set automatically the owner as responsible on owner set.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::ResponsibleAutoSet'} = '1';</code>

Ticket::Type

Description	Value
Description:	If ticket type feature is enabled, you can define ticket types for tickets (e. g. incident, problem, change, ...).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Type'} = '0';</code>

Ticket::Service

Description	Value
Description:	If ticket service/SLA feature is enabled, you can define ticket services and SLAs for tickets (e. g. email, desktop, network, ...).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Service'} = '0';</code>

Ticket::NumberGenerator

Description	Value
Description:	Select the ticket number generator module. "AutoIncrement" increments the ticket number, the SystemID and the counter

Description	Value
	are used. The format is SystemID.counter, e.g. 1010138, 1010139. With "Date" the ticket numbers will be generated by the current date, the SystemID and the counter. The format looks like Year.Month.Day.SystemID.counter, e.g. 200206231010138, 200206231010139. With "DateChecksum" the counter will be appended as 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:	<pre>\$Self->{'Ticket::NumberGenerator'} = 'Kernel::System::Ticket::Number::DateChecksum';</pre>

Ticket::NumberGenerator::MinCounterSize

Description	Value
Description:	If "AutoIncrement" was selected as TicketNumberGenerator you can set the minimal ticket counter size. Default is 5, this means the counter starts from 10000.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::NumberGenerator::MinCounterSize'} = '5';</pre>

Ticket::NumberGenerator::CheckSystemID

Description	Value
Description:	Check SystemID in ticket number detection for follow ups (use "No" if SystemID has been changed after using the system).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::NumberGenerator::CheckSystemID'} = '1';</pre>

Ticket::CounterLog

Description	Value
Description:	Logfile for the ticket counter.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::CounterLog'} = '<OTRS_CONFIG_Home>/var/log/TicketCounter.log';</pre>

Ticket::IndexModule

Description	Value
Description:	IndexAccelerator: choose your backend TicketViewAccelerator module. "RuntimeDB" generates each QueueView on the fly from ticket table. You won't have performance problems up to approx. 60.000 tickets in total and 6.000 open tickets in your system. "StaticDB" is the most powerful module. It should be used, if more than 80.000 and 6.000 open tickets are stored in the system. The module uses an extra ticket-index table that works like a view. Use the script "bin/RebuildTicketIndex.pl" for initial index update.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::IndexModule'} = 'Kernel::System::Ticket::IndexAccelerator::RuntimeDB';</pre>

Ticket::StorageModule

Description	Value
Description:	Select your TicketStorageModule to save the attachments of articles. "DB" stores all data in the database. Don't use this module if big attachments will be stored. "FS" stores the data in the filesystem. This is faster but webserver user should be the otrs user. You can switch between the modules even on a running system without any loss of data.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	<code>\$Self->{'Ticket::StorageModule'} = 'Kernel::System::Ticket::ArticleStorageDB';</code>

ArticleDir

Description	Value
Description:	If you have selected "FS" for TicketStorageModule, you have to specify the directory where to store the data.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'ArticleDir'} = '<OTRS_CONFIG_Home>/var/ article';</code>

Ticket::EventModulePost###1-AcceleratorUpdate

Description	Value
Description:	Ticket event module updates the ticket accelerator index.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::EventModulePost'}->{'1-AcceleratorUpdate'} = { 'Event' => '(TicketStateUpdate TicketQueueUpdate TicketLockUpdate)', 'Module' => 'Kernel::System::Ticket::Event::TicketAcceleratorUpdate' };</code>

Ticket::EventModulePost###1-ForceStateChangeOnLock

Description	Value
Description:	Ticket event module to force a new ticket state after lock action. As key you have to define the current state and the next state as content after lock action.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	<pre>\$Self->{'Ticket::EventModulePost'}->{'1-ForceStateChangeOnLock'} = { 'Event' => 'TicketLockUpdate', 'Module' => 'Kernel::System::Ticket::Event::ForceState', 'new' => 'open' };</pre>

Ticket::EventModulePost###1-ForceOwnerResetOnMove

Description	Value
Description:	Ticket event module to reset and unlock the owner if ticket gets moved to another queue.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}->{'1-ForceOwnerResetOnMove'} = { 'Event' => 'TicketQueueUpdate', 'Module' => 'Kernel::System::Ticket::Event::ForceOwnerReset' };</pre>

Ticket::EventModulePost###1-ResponsibleAutoSet

Description	Value
Description:	Ticket event module to set the responsible (if not already set) automatically on owner update after first owner update.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}->{'1-ResponsibleAutoSet'} = { 'Event' => 'TicketOwnerUpdate', 'Module' => 'Kernel::System::Ticket::Event::ResponsibleAutoSet' };</pre>

Ticket::EventModulePost###5-NotificationEvent

Description	Value
Description:	Ticket event module to send notifications.

Description	Value
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}->{'5-NotificationEvent'} = { 'Event' => '', 'Module' => 'Kernel::System::Ticket::Event::NotificationEvent', 'Transaction' => '1' };</pre>

Ticket::EventModulePost###99-ForceUnlockOnMove

Description	Value
Description:	Ticket event module to force tickets to be unlocked after moving to another queue.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}->{'99-ForceUnlockOnMove'} = { 'Event' => 'TicketQueueUpdate', 'Module' => 'Kernel::System::Ticket::Event::ForceUnlock' };</pre>

Ticket::EventModulePost###99-EscalationIndex

Description	Value
Description:	Ticket event module updates the ticket escalation index.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}->{'99-EscalationIndex'} = { 'Event' => '(TicketSLAUpdate TicketQueueUpdate TicketStateUpdate TicketCreate ArticleCreate)', 'Module' => 'Kernel::System::Ticket::Event::TicketEscalationIndex' };</pre>

Ticket::CustomModule

Description	Value
Description:	A module with custom functions to redefine Kernel::System::Ticket functions (e.g. Kernel::System::Ticket::Custom).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::CustomModule'} = 'Kernel::System::Ticket::Custom';</pre>

Ticket::ViewableSenderTypes

Description	Value
Description:	Default viewable sender types (default: customer).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::ViewableSenderTypes'} = ['\customer\'];</pre>

Ticket::ViewableLocks

Description	Value
Description:	Viewable locks. Default: unlock, tmp_lock.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::ViewableLocks'} = ['\unlock\'', '\tmp_lock\'];</pre>

Ticket::ViewableStateType

Description	Value
Description:	State types for a ticket to display.
Group:	Ticket

Description	Value
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::ViewableStateType'} = ['new', 'open', 'pending reminder', 'pending auto'];</pre>

Ticket::UnlockStateType

Description	Value
Description:	The states for unlocked tickets. You can unlock tickets with the script "bin/UnlockTickets.pl".
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::UnlockStateType'} = ['new', 'open'];</pre>

Ticket::PendingNotificationOnlyToOwner

Description	Value
Description:	Reminder notifications of unlocked ticket after reaching the reminder date only sent to ticket owner (not to whole agents of queue).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::PendingNotificationOnlyToOwner'} = '0';</pre>

Ticket::PendingNotificationNotToResponsible

Description	Value
Description:	Disable sending reminder notifications to responsible (Ticket::Responsible need to be activated for responsible support first).

Description	Value
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::PendingNotificationNotToResponsible'} = '0';</pre>

Ticket::PendingReminderStateType

Description	Value
Description:	State type of reminder for pending tickets.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::PendingReminderStateType'} = ['pending reminder'];</pre>

Ticket::PendingAutoStateType

Description	Value
Description:	Possible states for pending tickets that changed state after reached time limit.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::PendingAutoStateType'} = ['pending auto'];</pre>

Ticket::StateAfterPending

Description	Value
Description:	Define which State should be set automatically (Value) after pending time of State (Key) has been reached.
Group:	Ticket
SubGroup:	Core::Ticket

Description	Value
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::StateAfterPending'} = { 'pending auto close+' => 'closed successful', 'pending auto close-' => 'closed unsuccessful' };</pre>

System::Permission

Description	Value
Description:	These are the standard permissions available for agents within the application. If more permissions are needed, you can enter them here. Permissions must be hard coded to be effective. Some other good permissions have provisioning already built-in. The following permissions can also be used: note, close, pending, customer, freetext, move, compose, responsible, forward, and bounce. Please ensure, when adding any of the afore mentioned permissions, that the "rw" permission remains the last entry.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'System::Permission'} = ['ro', 'move_into', 'create', 'note', 'owner', 'priority', 'rw'];</pre>

Ticket::Permission###1-OwnerCheck

Description	Value
Description:	Module to check the owner.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Permission'}->{'1-OwnerCheck'} = { 'Granted' => '1',</pre>

Description	Value
	<pre>'Module' => 'Kernel::System::Ticket::Permission::OwnerCheck', 'Required' => '0' };</pre>

Ticket::Permission###2-ResponsibleCheck

Description	Value
Description:	Module to check the responsible.
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>

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>

Ticket::Permission###4-WatcherCheck

Description	Value
Description:	Module to check the watcher.
Group:	Ticket

Description	Value
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>

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>

CustomerTicket::Permission###2-CustomerUserIDCheck

Description	Value
Description:	Grant access, if customer ID is the same and group is accessible.
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>

Description	Value
	<code>};</code>

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>

CustomerNotifyJustToRealCustomer

Description	Value
Description:	Send customer notifications just to mapped customer. Normally if no customer is mapped, the latest customer sender gets the notification.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerNotifyJustToRealCustomer'} = '0';</pre>

Core::TicketACL

Ticket::Acl::Module###1-Ticket::Acl::Module

Description	Value
Description:	This ACL module allows you to not close parent tickets till all children's are closed ("State" shows which states should not be usable).
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	<pre>\$Self->{'Ticket::Acl::Module'}->{'1-Ticket::Acl::Module'} = { 'Module' => 'Kernel::System::Ticket::Acl::CloseParentAfterClosedChilds', 'State' => ['closed successful', 'closed unsuccessful'] };</pre>

TicketACL::Default::Action

Description	Value
Description:	Default ACL values for Ticket Actions.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TicketACL::Default::Action'} = {};</pre>

Core::TicketBulkAction

Ticket::Frontend::BulkFeature

Description	Value
Description:	Enable ticket bulk action feature for the agent frontend to work on more than one ticket at the same time.
Group:	Ticket
SubGroup:	Core::TicketBulkAction
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::BulkFeature'} = '1';</pre>

Ticket::Frontend::BulkFeatureGroup

Description	Value
Description:	Enable ticket bulk action feature only for the following groups.
Group:	Ticket
SubGroup:	Core::TicketBulkAction
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	<pre>\$Self->{'Ticket::Frontend::BulkFeatureGroup'} = ['admin', 'users'];</pre>

Core::TicketFreeText

TicketFreeKey1

Description	Value
Description:	Define the free key field 1 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey1'} = { 'Product' => 'Product' };</pre>

TicketFreeKey1::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 1 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey1::DefaultSelection'} = '';</pre>

TicketFreeText1

Description	Value
Description:	Define the FreeTextField 1 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	<pre>\$Self->{'TicketFreeText1'} = { '' => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

TicketFreeText1::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 1 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText1::DefaultSelection'} = 'Notebook';</pre>

TicketFreeText1::Link

Description	Value
Description:	Define the free text field 1 link for tickets. So this http link will be used in any ticket view.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText1::Link'} = 'http://some.example.com/handle?query= \$LQData{"TicketFreeText1"}';</pre>

TicketFreeKey2

Description	Value
Description:	Define the free key field 2 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	<pre>\$Self->{'TicketFreeKey2'} = { 'Product' => 'Product' };</pre>

TicketFreeKey2::DefaultSelection

Description	Value
Description:	The default selection of free key field 2 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey2::DefaultSelection'} = '';</pre>

TicketFreeText2

Description	Value
Description:	Define the FreeTextField 2 for tickets. It's a new ticket property.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText2'} = { '-' => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

TicketFreeText2::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 2 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText2::DefaultSelection'} = 'Notebook';</pre>

TicketFreeText2::Link

Description	Value
Description:	Define the free text field 2 link for tickets. So this http link will be used in any ticket view.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText2::Link'} = 'http://some.example.com/handle?query= \$LQData{"TicketFreeText2"}';</pre>

TicketFreeKey3

Description	Value
Description:	Define the free key field 3 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey3'} = { 'Product' => 'Product' };</pre>

TicketFreeKey3::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 3 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey3::DefaultSelection'} = '';</pre>

TicketFreeText3

Description	Value
Description:	Define the FreeTextField 3 for tickets. With this setting you can define a new ticket attribute.

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText3'} = { ' ' => ' ', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

TicketFreeText3::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 3 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText3::DefaultSelection'} = 'Notebook';</pre>

TicketFreeText3::Link

Description	Value
Description:	Define the free text field 3 link for tickets. So this http link will be used in any ticket view.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText3::Link'} = 'http://some.example.com/handle?query= \$LQData{"TicketFreeText3"}';</pre>

TicketFreeKey4

Description	Value
Description:	Define the free key field 4 for tickets. With this setting you can define a new ticket attribute.

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey4'} = { 'Product' => 'Product' };</pre>

TicketFreeKey4::DefaultSelection

Description	Value
Description:	The default selection of free key field 4 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey4::DefaultSelection'} = '';</pre>

TicketFreeText4

Description	Value
Description:	Define the FreeTextField 4 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText4'} = { '-' => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

TicketFreeText4::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 4 if more than one option is defined.
Group:	Ticket

Description	Value
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText4::DefaultSelection'} = 'Notebook';</pre>

TicketFreeText4::Link

Description	Value
Description:	Define the free text field 4 link for tickets. So this http link will be used in any ticket view.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText4::Link'} = 'http://some.example.com/handle?query= \$LQData{"TicketFreeText4"}';</pre>

TicketFreeKey5

Description	Value
Description:	Define the free key field 5 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey5'} = { 'Product' => 'Product' };</pre>

TicketFreeKey5::DefaultSelection

Description	Value
Description:	The default selection of free key field 5 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0

Description	Value
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey5::DefaultSelection'} = '';</code>

TicketFreeText5

Description	Value
Description:	Define the FreeTextField 5 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText5'} = { '-' => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

TicketFreeText5::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 5 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText5::DefaultSelection'} = 'Notebook';</pre>

TicketFreeText5::Link

Description	Value
Description:	Define the free text field 5 link for tickets. So this http link will be used in any ticket view.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'TicketFreeText5::Link'} = 'http://some.example.com/handle?query= \$LQData{"TicketFreeText5"}';</pre>

TicketFreeKey6

Description	Value
Description:	Define the free key field 6 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey6'} = { 'Product' => 'Product' };</pre>

TicketFreeKey6::DefaultSelection

Description	Value
Description:	The default selection of free key field 6 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey6::DefaultSelection'} = '';</pre>

TicketFreeText6

Description	Value
Description:	Define the FreeTextField 6 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText6'} = { '' => '-',</pre>

Description	Value
	<pre>'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

TicketFreeText6::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 6 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText6::DefaultSelection'} = 'Notebook';</pre>

TicketFreeText6::Link

Description	Value
Description:	Define the free text field 6 link for tickets. So this http link will be used in any ticket view.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText6::Link'} = 'http://some.example.com/handle?query= \$LQData{"TicketFreeText6"}';</pre>

TicketFreeKey7

Description	Value
Description:	Define the free key field 7 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey7'} = { 'Product' => 'Product'</pre>

Description	Value
	<code>};</code>

TicketFreeKey7::DefaultSelection

Description	Value
Description:	The default selection of free key field 7 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey7::DefaultSelection'} = '';</code>

TicketFreeText7

Description	Value
Description:	Define the FreeTextField 7 for tickets. With this setting you can define 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>

TicketFreeText7::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 7 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText7::DefaultSelection'} = 'Notebook';</code>

TicketFreeText7::Link

Description	Value
Description:	Define the free text field 7 link for tickets. So this http link will be used in any ticket view.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText7::Link'} = 'http://some.example.com/handle?query= \$LQData{"TicketFreeText7"}';</pre>

TicketFreeKey8

Description	Value
Description:	Define the free key field 8 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey8'} = { 'Product' => 'Product' };</pre>

TicketFreeKey8::DefaultSelection

Description	Value
Description:	The default selection of free key field 8 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey8::DefaultSelection'} = '';</pre>

TicketFreeText8

Description	Value
Description:	Define the FreeTextField 8 for tickets. With this setting you can define a new ticket attribute.

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText8'} = { ' ' => ' ', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

TicketFreeText8::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 8 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText8::DefaultSelection'} = 'Notebook';</pre>

TicketFreeText8::Link

Description	Value
Description:	Define the free text field 8 link for tickets. So this http link will be used in any ticket view.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText8::Link'} = 'http://some.example.com/handle?query= \$LQData{"TicketFreeText8"}';</pre>

TicketFreeKey9

Description	Value
Description:	Define the free key field 9 for tickets. With this setting you can define a new ticket attribute.

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey9'} = { 'Product' => 'Product' };</pre>

TicketFreeKey9::DefaultSelection

Description	Value
Description:	The default selection of free key field 9 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey9::DefaultSelection'} = '';</pre>

TicketFreeText9

Description	Value
Description:	Define the FreeTextField 9 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText9'} = { '-' => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

TicketFreeText9::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 9 if more than one option is defined.
Group:	Ticket

Description	Value
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText9::DefaultSelection'} = 'Notebook';</code>

TicketFreeText9::Link

Description	Value
Description:	Define the free text field 9 link for tickets. So this http link will be used in any ticket view.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText9::Link'} = 'http://some.example.com/handle?query=\$LQData{"TicketFreeText9"}';</code>

TicketFreeKey10

Description	Value
Description:	Define the free key field 10 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey10'} = { 'Product' => 'Product' };</code>

TicketFreeKey10::DefaultSelection

Description	Value
Description:	The default selection of free key field 10 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0

Description	Value
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey10::DefaultSelection'} = '';</code>

TicketFreeText10

Description	Value
Description:	Define the FreeTextField 10 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText10'} = { '' => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

TicketFreeText10::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 10 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText10::DefaultSelection'} = 'Notebook';</code>

TicketFreeText10::Link

Description	Value
Description:	Define the free text field 10 link for tickets. So this http link will be used in any ticket view.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'TicketFreeText10::Link'} = 'http://some.example.com/handle?query= \$LQData{"TicketFreeText10"}';</pre>

TicketFreeKey11

Description	Value
Description:	Define the free key field 11 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey11'} = { 'Product' => 'Product' };</pre>

TicketFreeKey11::DefaultSelection

Description	Value
Description:	The default selection of free key field 11 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey11::DefaultSelection'} = '';</pre>

TicketFreeText11

Description	Value
Description:	Define the FreeTextField 11 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText11'} = { '' => '-',</pre>

Description	Value
	<pre>'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

TicketFreeText11::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 11 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText11::DefaultSelection'} = 'Notebook';</pre>

TicketFreeText11::Link

Description	Value
Description:	Define the free text field 11 link for tickets. So this http link will be used in any ticket view.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText11::Link'} = 'http://some.example.com/handle?query= \$LQData{"TicketFreeText11"}';</pre>

TicketFreeKey12

Description	Value
Description:	Define the free key field 12 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey12'} = { 'Product' => 'Product'</pre>

Description	Value
	<code>};</code>

TicketFreeKey12::DefaultSelection

Description	Value
Description:	The default selection of free key field 12 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey12::DefaultSelection'} = '';</code>

TicketFreeText12

Description	Value
Description:	Define the FreeTextField 12 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText12'} = { ' ' => ' ', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</code>

TicketFreeText12::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 12 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText12::DefaultSelection'} = 'Notebook';</code>

TicketFreeText12::Link

Description	Value
Description:	Define the free text field 12 link for tickets. So this http link will be used in any ticket view.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText12::Link'} = 'http://some.example.com/handle?query= \$LQData{"TicketFreeText12"}';</pre>

TicketFreeKey13

Description	Value
Description:	Define the free key field 13 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey13'} = { 'Product' => 'Product' };</pre>

TicketFreeKey13::DefaultSelection

Description	Value
Description:	The default selection of free key field 13 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey13::DefaultSelection'} = '';</pre>

TicketFreeText13

Description	Value
Description:	Define the FreeTextField 13 for tickets. With this setting you can define a new ticket attribute.

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText13'} = { ' ' => ' ', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

TicketFreeText13::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 13 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText13::DefaultSelection'} = 'Notebook';</pre>

TicketFreeText13::Link

Description	Value
Description:	Define the free text field 13 link for tickets. So this http link will be used in any ticket view.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText13::Link'} = 'http://some.example.com/handle?query= \$LQData{"TicketFreeText13"}';</pre>

TicketFreeKey14

Description	Value
Description:	Define the free key field 14 for tickets. With this setting you can define a new ticket attribute.

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey14'} = { 'Product' => 'Product' };</pre>

TicketFreeKey14::DefaultSelection

Description	Value
Description:	The default selection of free key field 14 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey14::DefaultSelection'} = '';</pre>

TicketFreeText14

Description	Value
Description:	Define the FreeTextField 14 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText14'} = { '' => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

TicketFreeText14::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 14 if more than one option is defined.

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText14::DefaultSelection'} = 'Notebook';</pre>

TicketFreeText14::Link

Description	Value
Description:	Define the free text field 14 link for tickets. So this http link will be used in any ticket view.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText14::Link'} = 'http://some.example.com/handle?query= \$LQData{"TicketFreeText14"}';</pre>

TicketFreeKey15

Description	Value
Description:	Define the free key field 15 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey15'} = { 'Product' => 'Product' };</pre>

TicketFreeKey15::DefaultSelection

Description	Value
Description:	The default selection of free key field 15 if more than one option is defined.
Group:	Ticket

Description	Value
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeKey15::DefaultSelection'} = '';</code>

TicketFreeText15

Description	Value
Description:	Define the FreeTextField 15 for tickets. With this setting you can define 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>

TicketFreeText15::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 15 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeText15::DefaultSelection'} = 'Notebook';</code>

TicketFreeText15::Link

Description	Value
Description:	Define the free text field 15 link for tickets. So this http link will be used in any ticket view.
Group:	Ticket

Description	Value
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText15::Link'} = 'http://some.example.com/handle?query= \$LQData{"TicketFreeText15"}';</pre>

TicketFreeKey16

Description	Value
Description:	Define the free key field 16 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey16'} = { 'Product' => 'Product' };</pre>

TicketFreeKey16::DefaultSelection

Description	Value
Description:	The default selection of free key field 16 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeKey16::DefaultSelection'} = '';</pre>

TicketFreeText16

Description	Value
Description:	Define the FreeTextField 16 for tickets. With this setting you can define a new ticket attribute.
Group:	Ticket
SubGroup:	Core::TicketFreeText

Description	Value
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText16'} = { '' => '-', 'Notebook' => 'Notebook', 'PC' => 'PC', 'Phone' => 'Phone' };</pre>

TicketFreeText16::DefaultSelection

Description	Value
Description:	The default selection of FreeTextField 16 if more than one option is defined.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText16::DefaultSelection'} = 'Notebook';</pre>

TicketFreeText16::Link

Description	Value
Description:	Define the free text field 16 link for tickets. So this http link will be used in any ticket view.
Group:	Ticket
SubGroup:	Core::TicketFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeText16::Link'} = 'http://some.example.com/handle?query= \$LQData{"TicketFreeText16"}';</pre>

Core::TicketFreeTextDefault

Ticket::EventModulePost###TicketFreeFieldDefault

Description	Value
Description:	Registration of the TicketFreeFieldDefault event module. For more performance you can define a trigger event (e. g. Event =>

Description	Value
	TicketCreate). This is only possible if all TicketFreeField elements need the same event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::EventModulePost'}- >{'TicketFreeFieldDefault'} = { 'Module' => 'Kernel::System::Ticket::Event::TicketFreeFieldDefault', 'Transaction' => '1' };</pre>

Ticket::TicketFreeFieldDefault###Element1

Description	Value
Description:	Configure a default TicketFreeField setting. Counter defines the free text field you which should be used. With Key you set the TicketFreeKey and with Value the TicketFreeText. Event defines the trigger event. Please refer to the developer manual, chapter "Ticket Event Module" which can be found at http://doc.ots.org/ .
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}- >{'Element1'} = { 'Counter' => '1', 'Event' => 'TicketCreate', 'Key' => 'TicketType', 'Value' => 'Default' };</pre>

Ticket::TicketFreeFieldDefault###Element2

Description	Value
Description:	Configure a default TicketFreeField setting. Counter defines the free text field you which should be used. With Key you set the TicketFreeKey and with Value the TicketFreeText. Event defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0

Description	Value
Config-Setting:	<pre> \$Self->{'Ticket::TicketFreeFieldDefault'}- >{'Element2'} = { 'Counter' => '', 'Event' => '', 'Key' => '', 'Value' => '' }; </pre>

Ticket::TicketFreeFieldDefault###Element3

Description	Value
Description:	Configure a default TicketFreeField setting. Counter defines the free text field you which should be used. With Key you set the TicketFreeKey and with Value the TicketFreeText. 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>

Ticket::TicketFreeFieldDefault###Element4

Description	Value
Description:	Configure a default TicketFreeField setting. Counter defines the free text field you which should be used. With Key you set the TicketFreeKey and with Value the TicketFreeText. 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>

Ticket::TicketFreeFieldDefault###Element5

Description	Value
Description:	Configure a default TicketFreeField setting. Counter defines the free text field you which should be used. With Key you set the TicketFreeKey and with Value the TicketFreeText. 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>

Ticket::TicketFreeFieldDefault###Element6

Description	Value
Description:	Configure a default TicketFreeField setting. Counter defines the free text field you which should be used. With Key you set the TicketFreeKey and with Value the TicketFreeText. Event defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}- >{'Element6'} = { 'Counter' => '', 'Event' => '', 'Key' => '', 'Value' => '' };</pre>

Ticket::TicketFreeFieldDefault###Element7

Description	Value
Description:	Configure a default TicketFreeField setting. Counter defines the free text field you which should be used. With Key you set the TicketFreeKey and with Value the TicketFreeText. Event defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}- >{'Element7'} = { 'Counter' => '', 'Event' => '', 'Key' => '', 'Value' => '' };</pre>

Ticket::TicketFreeFieldDefault###Element8

Description	Value
Description:	Configure a default TicketFreeField setting. Counter defines the free text field you which should be used. With Key you set the TicketFreeKey and with Value the TicketFreeText. 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>

Ticket::TicketFreeFieldDefault###Element9

Description	Value
Description:	Configure a default TicketFreeField setting. Counter defines the free text field you which should be used. With Key you set the TicketFreeKey and with Value the TicketFreeText. 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>

Ticket::TicketFreeFieldDefault###Element10

Description	Value
Description:	Configure a default TicketFreeField setting. Counter defines the free text field you which should be used. With Key you set the TicketFreeKey and with Value the TicketFreeText. 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>

Ticket::TicketFreeFieldDefault###Element11

Description	Value
Description:	Configure a default TicketFreeField setting. Counter defines the free text field you which should be used. With Key you set the TicketFreeKey and with Value the TicketFreeText. Event defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'Ticket::TicketFreeFieldDefault'}- >{'Element11'} = { 'Counter' => '', 'Event' => '', 'Key' => '', 'Value' => '' }; </pre>

Ticket::TicketFreeFieldDefault###Element12

Description	Value
Description:	Configure a default TicketFreeField setting. Counter defines the free text field you which should be used. With Key you set the TicketFreeKey and with Value the TicketFreeText. Event defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0

Description	Value
Required:	0
Config-Setting:	<pre> \$Self->{'Ticket::TicketFreeFieldDefault'}- >{'Element12'} = { 'Counter' => '', 'Event' => '', 'Key' => '', 'Value' => '' }; </pre>

Ticket::TicketFreeFieldDefault###Element13

Description	Value
Description:	Configure a default TicketFreeField setting. Counter defines the free text field you which should be used. With Key you set the TicketFreeKey and with Value the TicketFreeText. 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>

Ticket::TicketFreeFieldDefault###Element14

Description	Value
Description:	Configure a default TicketFreeField setting. Counter defines the free text field you which should be used. With Key you set the TicketFreeKey and with Value the TicketFreeText. 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>

Description	Value
	<code>};</code>

Ticket::TicketFreeFieldDefault###Element15

Description	Value
Description:	Configure a default TicketFreeField setting. Counter defines the free text field you which should be used. With Key you set the TicketFreeKey and with Value the TicketFreeText. 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>

Ticket::TicketFreeFieldDefault###Element16

Description	Value
Description:	Configure a default TicketFreeField setting. Counter defines the free text field you which should be used. With Key you set the TicketFreeKey and with Value the TicketFreeText. Event defines the trigger event.
Group:	Ticket
SubGroup:	Core::TicketFreeTextDefault
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::TicketFreeFieldDefault'}- >{'Element16'} = { 'Counter' => '', 'Event' => '', 'Key' => '', 'Value' => '' };</pre>

Core::TicketFreeTime

TicketFreeTimeKey1

Description	Value
Description:	Define the key of TicketFreeTime field 1.

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeKey1'} = 'Termin1';</code>

TicketFreeTimeOptional1

Description	Value
Description:	Define, if TicketFreeTime field 1 is optional.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeOptional1'} = '1';</code>

TicketFreeTimeDiff1

Description	Value
Description:	Define the difference (in seconds) of the defaultvalue in TicketFreeTime field 1.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimeDiff1'} = '0';</code>

TicketFreeTimePeriod1

Description	Value
Description:	Define the period, which can selected in TicketFreeTime field 1.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimePeriod1'} = { 'YearPeriodFuture' => '5',</code>

Description	Value
	<code>'YearPeriodPast' => '5'</code> <code>};</code>

TicketFreeTimeKey2

Description	Value
Description:	Define the key of TicketFreeTime field 2.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeKey2'} = 'Termin2';</code>

TicketFreeTimeOptional2

Description	Value
Description:	Define, if TicketFreeTime field 2 is optional.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeOptional2'} = '1';</code>

TicketFreeTimeDiff2

Description	Value
Description:	Define the difference (in seconds) of the defaultvalue in TicketFreeTime field 2.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimeDiff2'} = '0';</code>

TicketFreeTimePeriod2

Description	Value
Description:	Define the period, which can selected in TicketFreeTime field 2.

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeTimePeriod2'} = { 'YearPeriodFuture' => '5', 'YearPeriodPast' => '5' };</pre>

TicketFreeTimeKey3

Description	Value
Description:	Define the key of TicketFreeTime field 3.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TicketFreeTimeKey3'} = 'Termin3';</pre>

TicketFreeTimeOptional3

Description	Value
Description:	Define, if TicketFreeTime field 3 is optional.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'TicketFreeTimeOptional3'} = '1';</pre>

TicketFreeTimeDiff3

Description	Value
Description:	Define the difference (in seconds) of the defaultvalue in TicketFreeTime field 3.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	<code>\$Self->{'TicketFreeTimeDiff3'} = '0';</code>

TicketFreeTimePeriod3

Description	Value
Description:	Define the period, which can selected in TicketFreeTime field 3.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'TicketFreeTimePeriod3'} = { 'YearPeriodFuture' => '5', 'YearPeriodPast' => '5' };</pre>

TicketFreeTimeKey4

Description	Value
Description:	Define the key of TicketFreeTime field 4.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeKey4'} = 'Termin4';</code>

TicketFreeTimeOptional4

Description	Value
Description:	Define, if TicketFreeTime field 4 is optional.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeOptional4'} = '1';</code>

TicketFreeTimeDiff4

Description	Value
Description:	Define the difference (in seconds) of the defaultvalue in TicketFreeTime field 4.

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimeDiff4'} = '0';</code>

TicketFreeTimePeriod4

Description	Value
Description:	Define the period, which can selected in TicketFreeTime field 4.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimePeriod4'} = { 'YearPeriodFuture' => '5', 'YearPeriodPast' => '5' };</code>

TicketFreeTimeKey5

Description	Value
Description:	Define the key of TicketFreeTime field 5.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeKey5'} = 'Termin5';</code>

TicketFreeTimeOptional5

Description	Value
Description:	Define, if TicketFreeTime field 5 is optional.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	<code>\$Self->{'TicketFreeTimeOptional5'} = '1';</code>

TicketFreeTimeDiff5

Description	Value
Description:	Define the difference (in seconds) of the defaultvalue in TicketFreeTime field 5.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimeDiff5'} = '0';</code>

TicketFreeTimePeriod5

Description	Value
Description:	Define the period, which can selected in TicketFreeTime field 5.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimePeriod5'} = { 'YearPeriodFuture' => '5', 'YearPeriodPast' => '5' };</code>

TicketFreeTimeKey6

Description	Value
Description:	Define the key of TicketFreeTime field 6.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeKey6'} = 'Termin6';</code>

TicketFreeTimeOptional6

Description	Value
Description:	Define, if TicketFreeTime field 6 is optional.

Description	Value
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'TicketFreeTimeOptional6'} = '1';</code>

TicketFreeTimeDiff6

Description	Value
Description:	Define the difference (in seconds) of the defaultvalue in TicketFreeTime field 6.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimeDiff6'} = '0';</code>

TicketFreeTimePeriod6

Description	Value
Description:	Define the period, which can selected in TicketFreeTime field 6.
Group:	Ticket
SubGroup:	Core::TicketFreeTime
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'TicketFreeTimePeriod6'} = { 'YearPeriodFuture' => '5', 'YearPeriodPast' => '5' };</code>

Core::TicketWatcher

Ticket::Watcher

Description	Value
Description:	Enable or disable the ticket watcher feature.
Group:	Ticket
SubGroup:	Core::TicketWatcher
Valid:	1

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

Ticket::WatcherGroup

Description	Value
Description:	Enable ticket watcher feature only for the following groups.
Group:	Ticket
SubGroup:	Core::TicketWatcher
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::WatcherGroup'} = ['admin', 'users'];</code>

Frontend::Admin::ModuleRegistration

Frontend::Module###AdminQueue

Description	Value
Description:	Frontend module registration for the AdminQueue object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Frontend::Module'}->{'AdminQueue'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block2', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Queue', 'Prio' => '100' }, 'NavBarName' => 'Admin', 'Title' => 'Queue' };</code>

Frontend::Module###AdminResponse

Description	Value
Description:	Frontend module registration for the AdminResponse object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminResponse'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block2', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Responses', 'Prio' => '200' }, 'NavBarName' => 'Admin', 'Title' => 'Response' }; </pre>

Frontend::Module###AdminQueueResponses

Description	Value
Description:	Frontend module registration for the AdminQueueResponses object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminQueueResponses'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block2', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Responses <-> Queue', 'Prio' => '300' } }; </pre>

Description	Value
	<pre> }, 'NavBarName' => 'Admin', 'Title' => 'Responses <-> Queue' }; </pre>

Frontend::Module###AdminAutoResponse

Description	Value
Description:	Frontend module registration for the AdminAutoResponses object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminAutoResponse'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block2', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Auto Responses', 'Prio' => '400' }, 'NavBarName' => 'Admin', 'Title' => 'Auto Responses' }; </pre>

Frontend::Module###AdminQueueAutoResponse

Description	Value
Description:	Frontend module registration for the AdminQueueAutoResponses object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}- >{'AdminQueueAutoResponse'} = { 'Description' => 'Admin', 'Group' => ['admin'], </pre>

Description	Value
	<pre> 'NavBarModule' => { 'Block' => 'Block2', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Auto Responses <-> Queue', 'Prio' => '500' }, 'NavBarName' => 'Admin', 'Title' => 'Auto Responses <-> Queue' }; </pre>

Frontend::Module###AdminAttachment

Description	Value
Description:	Frontend module registration for the AdminAttachment object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminAttachment'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block2', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Attachments', 'Prio' => '600' }, 'NavBarName' => 'Admin', 'Title' => 'Attachment' }; </pre>

Frontend::Module###AdminResponseAttachment

Description	Value
Description:	Frontend module registration for the AdminResponseAttachment object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	<pre> \$Self->{'Frontend::Module'}- >{'AdminResponseAttachment'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block2', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Attachments <-> Responses', 'Prio' => '700' }, 'NavBarName' => 'Admin', 'Title' => 'Attachments <-> Responses' }; </pre>

Frontend::Module###AdminSalutation

Description	Value
Description:	Frontend module registration for the AdminSalutation object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminSalutation'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Salutation', 'Prio' => '100' }, 'NavBarName' => 'Admin', 'Title' => 'Salutation' }; </pre>

Frontend::Module###AdminSignature

Description	Value
Description:	Frontend module registration for the AdminSignature object in the admin area.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminSignature'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Signature', 'Prio' => '200' }, 'NavBarName' => 'Admin', 'Title' => 'Signature' }; </pre>

Frontend::Module###AdminSystemAddress

Description	Value
Description:	Frontend module registration for the AdminSystemAddress object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminSystemAddress'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Email Addresses', 'Prio' => '300' }, 'NavBarName' => 'Admin', 'Title' => 'System address' }; </pre>

Frontend::Module###AdminNotification

Description	Value
Description:	Frontend module registration for the AdminNotification object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminNotification'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Notification', 'Prio' => '400' }, 'NavBarName' => 'Admin', 'Title' => 'Notification' }; </pre>

Frontend::Module###AdminNotificationEvent

Description	Value
Description:	Frontend module registration for the AdminNotificationEvent object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}- >{'AdminNotificationEvent'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Notification (Event)', </pre>

Description	Value
	<pre> 'Prio' => '400' }, 'NavBarName' => 'Admin', 'Title' => 'Notification' }; </pre>

Frontend::Module###AdminService

Description	Value
Description:	Frontend module registration for the AdminService configuration in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminService'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Service', 'Prio' => '900' }, 'NavBarName' => 'Admin', 'Title' => 'Service' }; </pre>

Frontend::Module###AdminSLA

Description	Value
Description:	Frontend module registration for the AdminSLA configuration in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminSLA'} = { 'Description' => 'Admin', 'Group' => ['admin'], </pre>

Description	Value
	<pre> 'NavBarModule' => { 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'SLA', 'Prio' => '1000' }, 'NavBarName' => 'Admin', 'Title' => 'SLA' }; </pre>

Frontend::Module###AdminType

Description	Value
Description:	Frontend module registration for the AdminType object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminType'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Type', 'Prio' => '700' }, 'NavBarName' => 'Admin', 'Title' => 'Type' }; </pre>

Frontend::Module###AdminState

Description	Value
Description:	Frontend module registration for the AdminState object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	<pre> \$Self->{'Frontend::Module'}->{'AdminState'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Status', 'Prio' => '800' }, 'NavBarName' => 'Admin', 'Title' => 'State' }; </pre>

Frontend::Module###AdminPriority

Description	Value
Description:	Frontend module registration for the AdminPriority object in the admin area.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminPriority'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block3', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Priority', 'Prio' => '850' }, 'NavBarName' => 'Admin', 'Title' => 'Priority' }; </pre>

Frontend::Module###AdminGenericAgent

Description	Value
Description:	Frontend module registration for the AdminGenericAgent object in the admin area.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AdminGenericAgent'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Block4', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'GenericAgent', 'Prio' => '300' }, 'NavBarName' => 'Admin', 'Title' => 'GenericAgent' }; </pre>

Frontend::Agent

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).
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::PendingDiffTime'} = '86400'; </pre>

Ticket::Frontend::ListType

Description	Value
Description:	Show parent/child queue lists in system as tree or as list.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::ListType'} = 'tree';</code>

Ticket::Frontend::StdResponsesMode

Description	Value
Description:	The standard responses in the agent interface can be displayed in a selection-box (form) or as links. Use Form if you have lots of responses.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::StdResponsesMode'} = 'Link';</code>

Ticket::Frontend::TextAreaEmail

Description	Value
Description:	Width of compose email windows.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::TextAreaEmail'} = '82';</code>

Ticket::Frontend::TextAreaNote

Description	Value
Description:	Width of compose note windows.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::TextAreaNote'} = '78';</code>

Ticket::Frontend::CustomerInfoCompose

Description	Value
Description:	Show customer user info in Compose screen (Phone and email)?

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

Ticket::Frontend::CustomerInfoComposeMaxSize

Description	Value
Description:	Max size (in characters) of customer info table - Compose screen (Phone and Email).
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerInfoComposeMaxSize'} = '22';</code>

Ticket::Frontend::CustomerInfoZoom

Description	Value
Description:	Show customer user info in ticket zoom view?
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerInfoZoom'} = '1';</code>

Ticket::Frontend::CustomerInfoZoomMaxSize

Description	Value
Description:	Max size (in characters) of customer info table - Zoom.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	<pre> \$Self->{'Ticket::Frontend::CustomerInfoZoomMaxSize'} = '22'; </pre>

Ticket::Frontend::CustomerInfoQueue

Description	Value
Description:	Show customer user info in QueueView?
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::CustomerInfoQueue'} = '0'; </pre>

Ticket::Frontend::CustomerInfoQueueMaxSize

Description	Value
Description:	Max size (in characters) of customer info table - Queue.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::CustomerInfoQueueMaxSize'} = '18'; </pre>

Ticket::Frontend::AccountTime

Description	Value
Description:	Activate time accounting.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::AccountTime'} = '1'; </pre>

Ticket::Frontend::TimeUnits

Description	Value
Description:	Your choice of your used time units (e.g. work units, hours, minutes).

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::TimeUnits'} = ' (work units) ';</code>

Ticket::Frontend::NeedAccountedTime

Description	Value
Description:	Does the time have to be accounted?
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NeedAccountedTime'} = '0';</code>

Ticket::Frontend::NeedSpellCheck

Description	Value
Description:	Do composed messages have to be spell checked?
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NeedSpellCheck'} = '0';</code>

Ticket::Frontend::NewOwnerSelection

Description	Value
Description:	If you want to show the owner selection in phone and email ticket.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::NewOwnerSelection'} = '1';</code>

Ticket::Frontend::NewResponsibleSelection

Description	Value
Description:	If you want to show the responsible selection in phone and email ticket.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::NewResponsibleSelection'} = '1';</pre>

Ticket::Frontend::NewQueueSelectionType

Description	Value
Description:	To: selection type. "Queue" shows all queues, "SystemAddress" displays all system addresses.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::NewQueueSelectionType'} = 'Queue';</pre>

Ticket::Frontend::NewQueueSelectionString

Description	Value
Description:	If "Queue" was selected for NewQueueSelectionType, you can specify a selection string. For example, "<Queue>" is useful for Queue and "<Realname> <<Email>>" for SystemAddress.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::NewQueueSelectionString'} = '<Queue>';</pre>

Ticket::Frontend::NewQueueOwnSelection

Description	Value
Description:	If this is in use, "just this selection is valid" for the PhoneView.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::NewQueueOwnSelection'} = { '1' => 'First Queue!', '2' => 'Second Queue!' };</pre>

Ticket::Frontend::ShowCustomerTickets

Description	Value
Description:	Do you want to show customer history tickets in AgentTicketPhone, AgentTicketEmail and AgentTicketCustomer?
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ShowCustomerTickets'} = '1';</pre>

CustomerDBLink

Description	Value
Description:	External link to customer database (e.g. 'http://yourhost/customer.php?CID=\$Data{"CustomerID"}' or "").
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerDBLink'} = '\$Env{"CGIHandle"}? Action=AgentTicketCustomer&TicketID= \$Data{"TicketID"}';</pre>

CustomerDBLinkTarget

Description	Value
Description:	Target attribute in Link to external customer DB. E.g. 'target="cdb"'.
Group:	Ticket

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

Frontend::CommonObject###QueueObject

Description	Value
Description:	Path to the file where all settings for the QueueObject object for the frontend are stored.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::CommonObject'}->{'QueueObject'} = 'Kernel::System::Queue';</code>

Frontend::CommonObject###TicketObject

Description	Value
Description:	Path to the file where all settings for the TicketObject object are stored.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::CommonObject'}->{'TicketObject'} = 'Kernel::System::Ticket';</code>

Frontend::CommonParam###Action

Description	Value
Description:	Default value for the Action parameter. The Frontend-Module which is used as default if no Action parameter given.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::CommonParam'}->{'Action'} = 'AgentDashboard';</code>

Frontend::CommonParam###QueueID

Description	Value
Description:	Default ID for queue, that is used by the system.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::CommonParam'}->{'QueueID'} = '0';</code>

Frontend::CommonParam###TicketID

Description	Value
Description:	Default ID for ticket, that is used by the system.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Frontend::CommonParam'}->{'TicketID'} = '';</code>

Frontend::Agent::CustomerSearch

Ticket::Frontend::CustomerSearchAutoComplete###Active

Description	Value
Description:	Enable or disable 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'} = '1';</code>

Ticket::Frontend::CustomerSearchAutoComplete###MinQueryLength

Description	Value
Description:	Set minimum number of characters before autocomplete query is sent.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::CustomerSearch
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::CustomerSearchAutoComplete'}- >{'MinQueryLength'} = '2';</pre>

Ticket::Frontend::CustomerSearchAutoComplete###QueryDelay

Description	Value
Description:	Delay time between autocomplete queries.
Group:	Ticket
SubGroup:	Frontend::Agent::CustomerSearch
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::CustomerSearchAutoComplete'}- >{'QueryDelay'} = '0.1';</pre>

Ticket::Frontend::CustomerSearchAutoComplete###TypeAhead

Description	Value
Description:	Enable or disable TypeAhead for the autocomplete feature.
Group:	Ticket
SubGroup:	Frontend::Agent::CustomerSearch
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::CustomerSearchAutoComplete'}- >{'TypeAhead'} = 'false';</pre>

Ticket::Frontend::CustomerSearchAutoComplete###MaxResultsDisplayed

Description	Value
Description:	Set number of search results for the autocomplete feature.
Group:	Ticket
SubGroup:	Frontend::Agent::CustomerSearch
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::CustomerSearchAutoComplete'}- >{'MaxResultsDisplayed'} = '20';</pre>

Ticket::Frontend::CustomerSearchAutoComplete::DynamicWidth

Description	Value
Description:	Should the search results container for the autocomplete feature dynamically adjust its width?
Group:	Ticket
SubGroup:	Frontend::Agent::CustomerSearch
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::CustomerSearchAutoComplete::DynamicWidth'} = '1';</pre>

Frontend::Agent::Dashboard

DashboardBackend###0100-TicketPendingReminder

Description	Value
Description:	Parameters for the dashboard backend. "Limit" are default shown entries. "Group" are used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" means if the plugin is enabled per default or if the user needs to enable it manually. "CacheTTLLocal" means the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'DashboardBackend'}->{'0100- TicketPendingReminder'} = { 'Attributes' => 'TicketPendingTimeOlderMinutes=1;StateType=pending reminder;SortBy=PendingTime;OrderBy=Down;', 'Block' => 'ContentLarge', 'CacheTTLLocal' => '0.5', 'Default' => '1', 'Description' => 'All tickets where the reminder date has reached!', 'Filter' => 'Locked', 'Group' => '', 'Limit' => '10',</pre>

Description	Value
	<pre>'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric', 'Permission' => 'rw', 'Time' => 'UntilTime', 'Title' => 'Reminder Tickets' };</pre>

DashboardBackend###0110-TicketEscalation

Description	Value
Description:	Parameters for the dashboard backend. "Limit" are default shown entries. "Group" are used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" means if the plugin is enabled per default or if the user needs to enable it manually. "CacheTTLocal" means the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'DashboardBackend'}->{'0110-TicketEscalation'} = { 'Attributes' => 'TicketEscalationTimeOlderMinutes=1;SortBy=EscalationTime;Order' 'Block' => 'ContentLarge', 'CacheTTLocal' => '0.5', 'Default' => '1', 'Description' => 'All tickets which are escalated!', 'Filter' => 'All', 'Group' => '', 'Limit' => '10', 'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric', 'Permission' => 'rw', 'Time' => 'EscalationTime', 'Title' => 'Escalated Tickets' };</pre>

DashboardBackend###0120-TicketNew

Description	Value
Description:	Parameters for the dashboard backend. "Limit" are default shown entries. "Group" are used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" means if the plugin is enabled per default or if the user needs to enable it manually. "CacheTTLocal" means 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'}->{'0120-TicketNew'} = { 'Attributes' => 'StateType=new;', 'Block' => 'ContentLarge', 'CacheTTLLocal' => '0.5', 'Default' => '1', 'Description' => 'All new tickets!', 'Filter' => 'All', 'Group' => '', 'Limit' => '10', 'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric', 'Permission' => 'rw', 'Time' => 'Age', 'Title' => 'New Tickets' }; </pre>

DashboardBackend###0130-TicketOpen

Description	Value
Description:	Parameters for the dashboard backend. "Limit" are default shown entries. "Group" are used to restriced access to the plugin (e. g. Group: admin;group1;group2;). "Default" means if the plugin is enabled per default or if the user needs to enable it manually. "CacheTTLLocal" means 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' => 'Tickets which need to be answered!', 'Filter' => 'All', 'Group' => '', 'Limit' => '10', 'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric', 'Permission' => 'rw', 'Time' => 'Age', }; </pre>

Description	Value
	<code>'Title' => 'Open Tickets / Need to be answered'</code> <code>};</code>

DashboardBackend###0250-TicketStats

Description	Value
Description:	Parameters for the dashboard backend. "Group" are used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" means if the plugin is enabled per default or if the user needs to enable it manually. "CacheTTL" means 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' => '45', 'Closed' => '1', 'Created' => '1', 'Default' => '1', 'Group' => '', 'Module' => 'Kernel::Output::HTML::DashboardTicketStatsGeneric', 'Permission' => 'rw', 'Title' => '7 Day Stats' };</pre>

DashboardBackend###0260-TicketCalendar

Description	Value
Description:	Parameters for the dashboard backend. "Limit" are default shown entries. "Group" are used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" means if the plugin is enabled per default or if the user needs to enable it manually. "CacheTTL" means 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',</pre>

Description	Value
	<pre>'Group' => '', 'Limit' => '6', 'Module' => 'Kernel::Output::HTML::DashboardCalendar', 'Permission' => 'rw', 'Title' => 'Upcoming Events' };</pre>

Frontend::Agent::ModuleNotify

Frontend::NotifyModule###3-Ticket::AgentTicketSeen

Description	Value
Description:	Modul for the agent interface that to sets tickets to "seen".
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::NotifyModule'}->{'3- Ticket::AgentTicketSeen'} = { 'Module' => 'Kernel::Output::HTML::NotificationAgentTicketSeen' };</pre>

Frontend::NotifyModule###4-Ticket::TicketNotify

Description	Value
Description:	Module to show agents important tickets via the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::NotifyModule'}->{'4- Ticket::TicketNotify'} = { 'Module' => 'Kernel::Output::HTML::NotificationAgentTicket ' };</pre>

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).

Description	Value
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>

Frontend::Agent::ModuleRegistration

Frontend::Module###AgentTicketQueue

Description	Value
Description:	Frontend module registration for the AgentTicketQueue object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketQueue'} = { 'Description' => 'Overview of all open Tickets', 'NavBar' => [{ 'AccessKey' => 'o', 'Block' => '', 'Description' => 'Overview of all open Tickets', 'Image' => 'overview.png', 'Link' => 'Action=AgentTicketQueue', 'Name' => 'QueueView', 'NavBar' => 'Ticket', 'Prio' => '100', 'Type' => '' }, { 'AccessKey' => 't', 'Block' => 'ItemArea', 'Description' => 'Ticket-Area', 'Image' => 'desktop.png', 'Link' => 'Action=AgentTicketQueue', </pre>

Description	Value
	<pre> 'Name' => 'Ticket', 'NavBar' => 'Ticket', 'Prio' => '200', 'Type' => 'Menu' }], 'NavBarName' => 'Ticket', 'Title' => 'QueueView' }; </pre>

Frontend::Module###AgentTicketPhone

Description	Value
Description:	Frontend module registration for the AgentTicketPhone object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketPhone'} = { 'Description' => 'Create new Phone Ticket', 'NavBar' => [{ 'AccessKey' => 'n', 'Block' => '', 'Description' => 'Create new Phone Ticket (Inbound)', 'Image' => 'phone-new.png', 'Link' => 'Action=AgentTicketPhone', 'Name' => 'Phone-Ticket', 'NavBar' => 'Ticket', 'Prio' => '200', 'Type' => '' }], 'NavBarName' => 'Ticket', 'Title' => 'Phone-Ticket' }; </pre>

Frontend::Module###AgentTicketPhoneOutbound

Description	Value
Description:	Frontend module registration for the AgentTicketPhoneOutbound object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration

Description	Value
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}- >{'AgentTicketPhoneOutbound'} = { 'Description' => 'Phone Call', 'NavBarName' => 'Ticket', 'Title' => 'Phone-Ticket' };</pre>

Frontend::Module###AgentTicketEmail

Description	Value
Description:	Frontend module registration for the AgentTicketEmail object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketEmail'} = { 'Description' => 'Create new email Ticket', 'NavBar' => [{ 'AccessKey' => 'e', 'Block' => '', 'Description' => 'Create new Email Ticket and send this out (Outbound)', 'Image' => 'mail_new.png', 'Link' => 'Action=AgentTicketEmail', 'Name' => 'Email-Ticket', 'NavBar' => 'Ticket', 'Prio' => '210', 'Type' => '' }], 'NavBarName' => 'Ticket', 'Title' => 'Email-Ticket' };</pre>

Frontend::Module###AgentTicketSearch

Description	Value
Description:	Frontend module registration for the AgentTicketSearch object in the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketSearch'} = { 'Description' => 'Search Tickets', 'NavBar' => [{ 'AccessKey' => 's', 'Block' => '', 'Description' => 'Search Tickets', 'Image' => 'search.png', 'Link' => 'Action=AgentTicketSearch', 'Name' => 'Search', 'NavBar' => 'Ticket', 'Prio' => '300', 'Type' => '' }], 'NavBarName' => 'Ticket', 'Title' => 'Search' }; </pre>

Frontend::Module###AgentTicketMailbox

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

Frontend::Module###AgentTicketLockedView

Description	Value
Description:	Frontend module registration for the AgentTicketLockedView object in the agent interface.

Description	Value
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>

Frontend::Module###AgentTicketResponsibleView

Description	Value
Description:	Frontend module registration for the AgentTicketResponsibleView object in 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>

Frontend::Module###AgentTicketWatchView

Description	Value
Description:	Frontend module registration for the AgentTicketWatchView object in 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>

Frontend::Module###AgentCustomerSearch

Description	Value
Description:	Frontend module registration for the AgentCustomerSearch object in 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>

Frontend::Module###AgentTicketStatusView

Description	Value
Description:	Frontend module registration for the AgentTicketStatusView object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	0
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', 'Image' => 'overview.png', 'Link' => 'Action=AgentTicketStatusView', 'Name' => 'StatusView', 'NavBar' => 'Ticket', 'Prio' => '110', 'Type' => '' }], 'NavBarName' => 'Ticket', 'Title' => 'Status View' };</pre>

Frontend::Module###AgentTicketEscalationView

Description	Value
Description:	Frontend module registration for the AgentTicketEscalationView object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}->{'AgentTicketEscalationView'} = { 'Description' => 'Overview of all escalated tickets', 'NavBar' => [{ 'AccessKey' => 'e', 'Block' => '', 'Description' => 'Overview Escalated Tickets', 'Image' => 'overview.png', 'Link' => 'Action=AgentTicketEscalationView', 'Name' => 'EscalationView', 'NavBar' => 'Ticket', 'Prio' => '120', 'Type' => '' }], 'NavBarName' => 'Ticket', 'Title' => 'Escalation View' }; </pre>

Frontend::Module###AgentZoom

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

Frontend::Module###AgentTicketZoom

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

Frontend::Module###AgentTicketAttachment

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

Frontend::Module###AgentTicketPlain

Description	Value
Description:	Frontend module registration for the AgentTicketPlain object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	<pre>\$Self->{'Frontend::Module'}->{'AgentTicketPlain'} = { 'Description' => 'Ticket plain view of an email', 'NavBarName' => 'Ticket', 'Title' => 'Plain' };</pre>

Frontend::Module###AgentTicketNote

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

Frontend::Module###AgentTicketMerge

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

Frontend::Module###AgentTicketPending

Description	Value
Description:	Frontend module registration for the AgentTicketPending object in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}- >{'AgentTicketPending'} = { 'Description' => 'Ticket Pending', 'NavBarName' => 'Ticket', 'Title' => 'Pending' }; </pre>

Frontend::Module###AgentTicketWatcher

Description	Value
Description:	All parameters for the TicketWatcher object in 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>

Frontend::Module###AgentTicketPriority

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

Frontend::Module###AgentTicketLock

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

Frontend::Module###AgentTicketMove

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

Frontend::Module###AgentTicketHistory

Description	Value
Description:	Frontend module registration for the AgentTicketHistory object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	<pre>\$Self->{'Frontend::Module'}- >{'AgentTicketHistory'} = { 'Description' => 'Ticket History', 'NavBarName' => 'Ticket', 'Title' => 'History' };</pre>

Frontend::Module###AgentTicketOwner

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

Frontend::Module###AgentTicketResponsible

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

Frontend::Module###AgentTicketCompose

Description	Value
Description:	Frontend module registration for the AgentTicketCompose object in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::Module'}- >{'AgentTicketCompose'} = { 'Description' => 'Ticket Compose email Answer', 'NavBarName' => 'Ticket', 'Title' => 'Compose' }; </pre>

Frontend::Module###AgentTicketBounce

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

Frontend::Module###AgentTicketForward

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

Frontend::Module###AgentTicketCustomer

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

Frontend::Module###AgentTicketClose

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

Frontend::Module###AgentTicketFreeText

Description	Value
Description:	Frontend module registration for the AgentTicketFreeText object in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::Module'}- >{'AgentTicketFreeText'} = {</pre>

Description	Value
	<pre>'Description' => 'Ticket FreeText', 'NavBarName' => 'Ticket', 'Title' => 'Free Fields' };</pre>

Frontend::Module###AgentTicketPrint

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

Frontend::Module###AgentTicketBulk

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

Frontend::Agent::NavBarModule

Frontend::NavBarModule###1-Ticket::TicketWatcher

Description	Value
Description:	Agent interface notification module to see the number of watched tickets.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::NavBarModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::NavBarModule'}->{'1-Ticket::TicketWatcher'} = { 'Module' => 'Kernel::Output::HTML::NavBarTicketWatcher' };</pre>

Frontend::NavBarModule###1-Ticket::LockedTickets

Description	Value
Description:	Agent interface notification module to check the used charset.
Group:	Ticket
SubGroup:	Frontend::Agent::NavBarModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::NavBarModule'}->{'1-Ticket::LockedTickets'} = { 'Module' => 'Kernel::Output::HTML::NavBarLockedTickets' };</pre>

Frontend::NavBarModule###1-Ticket::TicketSearchProfile

Description	Value
Description:	Agent interface module to access search profiles via nav bar.
Group:	Ticket
SubGroup:	Frontend::Agent::NavBarModule
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Frontend::NavBarModule'}->{'1-Ticket::TicketSearchProfile'} = { 'Block' => 'ItemSearchProfile', 'Description' => 'Search-Template', 'MaxWidth' => '40', 'Module' => 'Kernel::Output::HTML::NavBarTicketSearchProfile', 'Name' => 'Search-Template' };</pre>

Frontend::NavBarModule###1-Ticket::TicketSearchFulltext

Description	Value
Description:	Agent interface module to access fulltext search via nav bar.
Group:	Ticket
SubGroup:	Frontend::Agent::NavBarModule
Valid:	0
Required:	0
Config-Setting:	<pre> \$Self->{'Frontend::NavBarModule'}->{'1-Ticket::TicketSearchFulltext'} = { 'Block' => 'ItemSearchFulltext', 'Description' => 'Fulltext-Search', 'Module' => 'Kernel::Output::HTML::NavBarTicketSearchFulltext', 'Name' => 'Fulltext-Search', 'Size' => '10' }; </pre>

Frontend::Agent::Preferences

PreferencesGroups###NewTicketNotify

Description	Value
Description:	Parameters for the NewTicketNotify object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'NewTicketNotify'} = { 'Activ' => '1', 'Colum' => 'Mail Management', 'Data' => { '0' => 'No', '1' => 'Yes' }, 'DataSelected' => '0', 'Desc' => 'Send me a notification if there is a new ticket in "My Queues".', 'Label' => 'New ticket notification', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserSendNewTicketNotification', 'Prio' => '1000' }; </pre>

PreferencesGroups###FollowUpNotify

Description	Value
Description:	Parameters for the FollowUpNotify object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'FollowUpNotify'} = { 'Activ' => '1', 'Colum' => 'Mail Management', 'Data' => { '0' => 'No', '1' => 'Yes' }, 'DataSelected' => '0', 'Desc' => 'Send me a notification if a customer sends a follow up and I\'m the owner of the ticket or the ticket is unlocked and is in one of my subscribed queues.', 'Label' => 'Follow up notification', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserSendFollowUpNotification', 'Prio' => '2000' };</pre>

PreferencesGroups###LockTimeoutNotify

Description	Value
Description:	Parameters for the LockTimeoutNotify object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'LockTimeoutNotify'} = { 'Activ' => '1', 'Colum' => 'Mail Management', 'Data' => { '0' => 'No', '1' => 'Yes' }, 'DataSelected' => '0', 'Desc' => 'Send me a notification if a ticket is unlocked by the system.',</pre>

Description	Value
	<pre>'Label' => 'Ticket lock timeout notification', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserSendLockTimeoutNotification', 'Prio' => '3000' };</pre>

PreferencesGroups###MoveNotify

Description	Value
Description:	Parameters for the MoveNotify object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'MoveNotify'} = { 'Activ' => '1', 'Colum' => 'Mail Management', 'Data' => { '0' => 'No', '1' => 'Yes' }, 'DataSelected' => '0', 'Desc' => 'Send me a notification if a ticket is moved into one of "My Queues".', 'Label' => 'Move notification', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserSendMoveNotification', 'Prio' => '4000' };</pre>

PreferencesGroups###WatcherNotify

Description	Value
Description:	Parameters for the WatcherNotify object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'PreferencesGroups'}->{'WatcherNotify'} = { 'Activ' => '1', 'Colum' => 'Mail Management', 'Data' => {</pre>

Description	Value
	<pre> '0' => 'No', '1' => 'Yes' }, 'DataSelected' => '0', 'Desc' => 'Send me a notification of an watched ticket like an owner of an ticket.', 'Label' => 'Watch notification', 'Module' => 'Kernel::Output::HTML::PreferencesTicketWatcher', 'PrefKey' => 'UserSendWatcherNotification', 'Prio' => '5000' }; </pre>

PreferencesGroups###CustomQueue

Description	Value
Description:	Parameters for the CustomQueue object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'CustomQueue'} = { 'Activ' => '1', 'Colum' => 'Other Options', 'Desc' => 'Your queue selection of your favourite queues. You also get notified about those queues via email if enabled.', 'Label' => 'My Queues', 'Module' => 'Kernel::Output::HTML::PreferencesCustomQueue', 'Permission' => 'ro', 'Prio' => '2000' }; </pre>

PreferencesGroups###RefreshTime

Description	Value
Description:	Parameters for the RefreshTime object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'RefreshTime'} = { 'Activ' => '1', 'Colum' => 'Frontend', 'Data' => { </pre>

Description	Value
	<pre> '0' => 'off', '10' => '10 minutes', '15' => '15 minutes', '2' => ' 2 minutes', '5' => ' 5 minutes', '7' => ' 7 minutes' }, 'DataSelected' => '0', 'Desc' => 'Select your QueueView refresh time.', 'Label' => 'QueueView refresh time', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserRefreshTime', 'Prio' => '3000' }; </pre>

PreferencesGroups###CreateNextMask

Description	Value
Description:	Parameters for the CreateNextMask object in the preference view.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'PreferencesGroups'}->{'CreateNextMask'} = { 'Activ' => '1', 'Colum' => 'Frontend', 'Data' => { '' => 'CreateTicket', 'AgentTicketZoom' => 'TicketZoom' }, 'DataSelected' => '', 'Desc' => 'Select your screen after creating a new ticket.', 'Label' => 'Screen after new ticket', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserCreateNextMask', 'Prio' => '5000' }; </pre>

Frontend::Agent::Ticket::ArticleAttachmentModule

Ticket::Frontend::ArticleAttachmentModule###1-Download

Description	Value
Description:	Links in agent zoom for attachments to download.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleAttachmentModule
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::ArticleAttachmentModule'}- >{'1-Download'} = { 'Module' => 'Kernel::Output::HTML::ArticleAttachmentDownload' };</pre>

Ticket::Frontend::ArticleAttachmentModule###2-HTML-Viewer

Description	Value
Description:	Links in agent zoom for attachments html online viewer.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleAttachmentModule
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::ArticleAttachmentModule'}- >{'2-HTML-Viewer'} = { 'Module' => 'Kernel::Output::HTML::ArticleAttachmentHTMLViewer' };</pre>

Frontend::Agent::Ticket::ArticleComposeModule

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>

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>

Frontend::Agent::Ticket::ArticleViewModule

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>

Ticket::Frontend::ArticleViewModule###1-SMIME

Description	Value
Description:	A 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
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ArticleViewModule'}->{'1-SMIME'} = {</pre>

Description	Value
	<pre>'Module' => 'Kernel::Output::HTML::ArticleCheckSMIME' };</pre>

Frontend::Agent::Ticket::ArticleViewModulePre

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>

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>

Frontend::Agent::Ticket::MenuModule

Ticket::Frontend::MenuModule###000-Back

Description	Value
Description:	Module to show back link in menu.
Group:	Ticket

Description	Value
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' }; </pre>

Ticket::Frontend::MenuModule###100-Lock

Description	Value
Description:	Module to show lock/unlock link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::MenuModule'}->{'100-Lock'} = { 'Action' => 'AgentTicketLock', 'Module' => 'Kernel::Output::HTML::TicketMenuLock', 'Name' => 'Lock' }; </pre>

Ticket::Frontend::MenuModule###200-History

Description	Value
Description:	Module to show history link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::MenuModule'}->{'200-History'} = { 'Action' => 'AgentTicketHistory', 'Description' => 'Shows the ticket history!', </pre>

Description	Value
	<pre>'Link' => 'Action=AgentTicketHistory&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'History' };</pre>

Ticket::Frontend::MenuModule###210-Print

Description	Value
Description:	Module to show print link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'210-Print'} = { 'Action' => 'AgentTicketPrint', 'Description' => 'Print this ticket!', 'Link' => 'Action=AgentTicketPrint&TicketID=\$QData{"TicketID"}', 'LinkParam' => 'target="print"', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Print' };</pre>

Ticket::Frontend::MenuModule###300-Priority

Description	Value
Description:	Module to show priority link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::MenuModule'}->{'300-Priority'} = { 'Action' => 'AgentTicketPriority', 'Description' => 'Change the ticket priority!', 'Link' => 'Action=AgentTicketPriority&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Priority' };</pre>

Ticket::Frontend::MenuModule###310-FreeText

Description	Value
Description:	Module to show free text link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::MenuModule'}->{'310-FreeText'} = { 'Action' => 'AgentTicketFreeText', 'Description' => 'Change the ticket free fields!', 'Link' => 'Action=AgentTicketFreeText&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Free Fields' }; </pre>

Ticket::Frontend::MenuModule###320-Link

Description	Value
Description:	Module to show a link to link a ticket with another object.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::MenuModule'}->{'320-Link'} = { 'Action' => 'AgentLinkObject', 'Description' => 'Link this ticket to an other objects!', 'Link' => 'Action=AgentLinkObject&SourceObject=Ticket&SourceKey=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Link' }; </pre>

Ticket::Frontend::MenuModule###400-Owner

Description	Value
Description:	Module to show owner link in menu.

Description	Value
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' }; </pre>

Ticket::Frontend::MenuModule###410-Responsible

Description	Value
Description:	Module to show responsible link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
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' }; </pre>

Ticket::Frontend::MenuModule###420-Customer

Description	Value
Description:	Module to show customer link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Description	Value
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' }; </pre>

Ticket::Frontend::MenuModule###420-Note

Description	Value
Description:	Module to show link to note in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::MenuModule'}->{'420- Note'} = { 'Action' => 'AgentTicketNote', 'Description' => 'Add a note to this ticket!', 'Link' => 'Action=AgentTicketNote&TicketID= \$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Note' }; </pre>

Ticket::Frontend::MenuModule###430-Merge

Description	Value
Description:	Module to show merge link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::MenuModule'}->{'430- Merge'} = { 'Action' => 'AgentTicketMerge', 'Description' => 'Merge this ticket!', 'Link' => 'Action=AgentTicketMerge&TicketID= \$QData{"TicketID"}', </pre>

Description	Value
	<pre>'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Merge' };</pre>

Ticket::Frontend::MenuModule###440-Pending

Description	Value
Description:	Module to show pending link in menu.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
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' };</pre>

Ticket::Frontend::MenuModule###448-Watch

Description	Value
Description:	Module to show subscribe/unsubscribe link in menu.
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' };</pre>

Ticket::Frontend::MenuModule###450-Close

Description	Value
Description:	Module to show close link in menu.

Description	Value
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' }; </pre>

Ticket::Frontend::MenuModule###460-Delete

Description	Value
Description:	Show delete link. 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=Delete', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Delete' }; </pre>

Ticket::Frontend::MenuModule###470-Spam

Description	Value
Description:	Show spam link. 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

Description	Value
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=Delete', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Spam' };</pre>

Frontend::Agent::Ticket::MenuModulePre

Ticket::Frontend::PreMenuModule###100-Lock

Description	Value
Description:	Show lock / unlock link.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::PreMenuModule'}->{'100-Lock'} = { 'Action' => 'AgentTicketLock', 'Module' => 'Kernel::Output::HTML::TicketMenuLock', 'Name' => 'Lock' };</pre>

Ticket::Frontend::PreMenuModule###200-Zoom

Description	Value
Description:	Show zoom link.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::PreMenuModule'}->{'200-Zoom'} = { 'Action' => 'AgentTicketZoom',</pre>

Description	Value
	<pre>'Description' => 'Look into a ticket!', 'Link' => 'Action=AgentTicketZoom&TicketID= \$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Zoom' };</pre>

Ticket::Frontend::PreMenuModule###210-History

Description	Value
Description:	Show history link.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::PreMenuModule'}->{'210- History'} = { 'Action' => 'AgentTicketHistory', 'Description' => 'Shows the ticket history!', 'Link' => 'Action=AgentTicketHistory&TicketID= \$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'History' };</pre>

Ticket::Frontend::PreMenuModule###300-Priority

Description	Value
Description:	Show priority link.
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' };</pre>

Ticket::Frontend::PreMenuModule###420-Note

Description	Value
Description:	Show note link.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::PreMenuModule'}->{'420-Note'} = { 'Action' => 'AgentTicketNote', 'Description' => 'Add a note to this ticket!', 'Link' => 'Action=AgentTicketNote&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Note' }; </pre>

Ticket::Frontend::PreMenuModule###440-Close

Description	Value
Description:	Show close link.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::PreMenuModule'}->{'440-Close'} = { 'Action' => 'AgentTicketClose', 'Description' => 'Close this ticket!', 'Link' => 'Action=AgentTicketClose&TicketID=\$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Close' }; </pre>

Ticket::Frontend::PreMenuModule###450-Delete

Description	Value
Description:	Show delete link. 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

Description	Value
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=Delete', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Delete' };</pre>

Ticket::Frontend::PreMenuModule###460-Spam

Description	Value
Description:	Show spam link. 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'}->{'460-Spam'} = { 'Action' => 'AgentTicketMove', 'Description' => 'Mark as Spam!', 'Link' => 'Action=AgentTicketMove&TicketID=\$Data{"TicketID"}&DestQueue=Delete', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Spam' };</pre>

Frontend::Agent::Ticket::ViewBounce

Ticket::Frontend::AgentTicketBounce###Permission

Description	Value
Description:	Required permissions to use this option.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1

Description	Value
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'Permission'} = 'bounce';</pre>

Ticket::Frontend::AgentTicketBounce###RequiredLock

Description	Value
Description:	A ticket lock is required. In case the ticket isn't locked, the tickets get locked and the current agent will be set automatically as ticket owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'RequiredLock'} = '1';</pre>

Ticket::Frontend::AgentTicketBounce###StateDefault

Description	Value
Description:	Default next state after bouncing a ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'StateDefault'} = 'closed successful';</pre>

Ticket::Frontend::AgentTicketBounce###StateType

Description	Value
Description:	Next possible states when a ticket is bounced.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'StateType'} = ['open', 'closed'];</pre>

Ticket::Frontend::BounceText

Description	Value
Description:	Default bounce notification for customer/sender.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::BounceText'} = 'Your email with ticket number "<OTRS_TICKET>" is bounced to "<OTRS_BOUNCE_TO>". Contact this address for further information.';</pre>

Frontend::Agent::Ticket::ViewBulk

Ticket::Frontend::AgentTicketBulk###Owner

Description	Value
Description:	If you want to set the owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Owner'} = '1';</pre>

Ticket::Frontend::AgentTicketBulk###Responsible

Description	Value
Description:	If you want to set the responsible.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Responsible'} = '1';</pre>

Ticket::Frontend::AgentTicketBulk###State

Description	Value
Description:	Would you like to set the state of a ticket if a note is created by an agent?

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketBulk'}- >{'State'} = '1';</pre>

Ticket::Frontend::AgentTicketBulk###StateType

Description	Value
Description:	Default next states after adding a note.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketBulk'}- >{'StateType'} = ['open', 'closed', 'pending reminder', 'pending auto'];</pre>

Ticket::Frontend::AgentTicketBulk###StateDefault

Description	Value
Description:	Default next state.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketBulk'}- >{'StateDefault'} = 'open';</pre>

Ticket::Frontend::AgentTicketBulk###Priority

Description	Value
Description:	Show priority options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Priority'} = '1';</pre>

Ticket::Frontend::AgentTicketBulk###PriorityDefault

Description	Value
Description:	Default priority options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'PriorityDefault'} = '3 normal';</pre>

Ticket::Frontend::AgentTicketBulk###ArticleTypeDefault

Description	Value
Description:	Default note type.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'ArticleTypeDefault'} = 'note-internal';</pre>

Ticket::Frontend::AgentTicketBulk###ArticleTypes

Description	Value
Description:	Specify the different note types, that you want to use in your 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>

Frontend::Agent::Ticket::ViewClose

Ticket::Frontend::AgentTicketClose###Permission

Description	Value
Description:	Required permissions to use this option.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Permission'} = 'close';</pre>

Ticket::Frontend::AgentTicketClose###RequiredLock

Description	Value
Description:	A ticket lock is required. In case the ticket isn't locked, the tickets get locked and the current agent will be set automatically as ticket owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'RequiredLock'} = '1';</pre>

Ticket::Frontend::AgentTicketClose###TicketType

Description	Value
Description:	If you want to set the ticket type (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'TicketType'} = '0';</pre>

Ticket::Frontend::AgentTicketClose###Service

Description	Value
Description:	If you want to set the service (Ticket::Service needs to be activated).
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Service'} = '0';</code>

Ticket::Frontend::AgentTicketClose###Owner

Description	Value
Description:	If you want to set the owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Owner'} = '0';</code>

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>

Ticket::Frontend::AgentTicketClose###Responsible

Description	Value
Description:	If you want to set the responsible.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Responsible'} = '0';</code>

Ticket::Frontend::AgentTicketClose###State

Description	Value
Description:	Would you like to set the state of a ticket if a note is created by an agent?
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'State'} = '1';</pre>

Ticket::Frontend::AgentTicketClose###StateType

Description	Value
Description:	Default next states after adding a note.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'StateType'} = ['closed'];</pre>

Ticket::Frontend::AgentTicketClose###StateDefault

Description	Value
Description:	Default next state.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'StateDefault'} = 'closed successful';</pre>

Ticket::Frontend::AgentTicketClose###Note

Description	Value
Description:	Show note fields.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}- >{'Note'} = '1';</pre>

Ticket::Frontend::AgentTicketClose###Subject

Description	Value
Description:	Default note subject.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}- >{'Subject'} = 'Text{"Close"}';</pre>

Ticket::Frontend::AgentTicketClose###Body

Description	Value
Description:	Default note text.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}- >{'Body'} = '';</pre>

Ticket::Frontend::AgentTicketClose###InvolvedAgent

Description	Value
Description:	Show selection of involved agents.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}- >{'InvolvedAgent'} = '0';</pre>

Ticket::Frontend::AgentTicketClose###InformAgent

Description	Value
Description:	Show selection of agents to inform (all agents with note permissions on the queue/ticket).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'InformAgent'} = '0';</pre>

Ticket::Frontend::AgentTicketClose###ArticleTypeDefault

Description	Value
Description:	Default note type.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ArticleTypeDefault'} = 'note-internal';</pre>

Ticket::Frontend::AgentTicketClose###ArticleTypes

Description	Value
Description:	Specify the different note types, that you want to use in your system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };</pre>

Ticket::Frontend::AgentTicketClose###Priority

Description	Value
Description:	Show priority options.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Priority'} = '0';</pre>

Ticket::Frontend::AgentTicketClose###PriorityDefault

Description	Value
Description:	Default priority options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'PriorityDefault'} = '3 normal';</pre>

Ticket::Frontend::AgentTicketClose###Title

Description	Value
Description:	Show title fields.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Title'} = '0';</pre>

Ticket::Frontend::AgentTicketClose###TicketFreeText

Description	Value
Description:	Shown ticket free text options. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	<pre> \$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' }; </pre>

Ticket::Frontend::AgentTicketClose###TicketFreeTime

Description	Value
Description:	Shown ticket free time options. 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'}- >{'TicketFreeTime'} = { '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' }; </pre>

Ticket::Frontend::AgentTicketClose###ArticleFreeText

Description	Value
Description:	Shown article free text options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}- >{'ArticleFreeText'} = { '1' => '0', '2' => '0', '3' => '0' };</pre>

Ticket::Frontend::AgentTicketClose###HistoryType

Description	Value
Description:	History type for this action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}- >{'HistoryType'} = 'AddNote';</pre>

Ticket::Frontend::AgentTicketClose###HistoryComment

Description	Value
Description:	History comment for this action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketClose'}- >{'HistoryComment'} = '%%Close';</pre>

Frontend::Agent::Ticket::ViewCompose

Ticket::Frontend::AgentTicketCompose###Permission

Description	Value
Description:	Required permissions to use this option.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	<code>\$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'Permission'} = 'compose';</code>

Ticket::Frontend::AgentTicketCompose###RequiredLock

Description	Value
Description:	A ticket lock is required. In case the ticket isn't locked, the tickets get locked and the current agent will be set automatically as ticket owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'RequiredLock'} = '1';</code>

Ticket::Frontend::AgentTicketCompose###StateDefault

Description	Value
Description:	Default next state if a ticket is composed, answered, e.g.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'StateDefault'} = 'open';</code>

Ticket::Frontend::AgentTicketCompose###StateType

Description	Value
Description:	Next possible states after composing / answering a ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'StateType'} = ['open', 'closed', 'pending auto', 'pending reminder'];</code>

Ticket::Frontend::ResponseFormat

Description	Value
Description:	Format of responses (\$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"} \$TimeShort{"\$QData{"Created"}} - \$QData{"OrigFromName"} \$Text{"wrote"}: \$QData{"Body"} \$QData{"StdResponse"} \$QData{"Signature"} '; </pre>

Ticket::Frontend::Quote

Description	Value
Description:	Character for email quotes.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::Quote'} = '>'; </pre>

Ticket::Frontend::ComposeAddCustomerAddress

Description	Value
Description:	Always add customers email address to recipients on compose answer screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self- >{'Ticket::Frontend::ComposeAddCustomerAddress'} = '1'; </pre>

Ticket::Frontend::ComposeReplaceSenderAddress

Description	Value
Description:	Replace on compose answer the original sender with current customers email address.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ComposeReplaceSenderAddress'} = '0';</pre>

Ticket::Frontend::ComposeExcludeCcRecipients

Description	Value
Description:	Do not use Cc recipients in reply Cc list on compose an email answer.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::ComposeExcludeCcRecipients'} = '0';</pre>

Ticket::Frontend::AgentTicketCompose###TicketFreeText

Description	Value
Description:	Shown ticket free text options. 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', }</pre>

Description	Value
	<pre>'15' => '0', '16' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' };</pre>

Ticket::Frontend::AgentTicketCompose###TicketFreeTime

Description	Value
Description:	Shown ticket free time options. 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>

Ticket::Frontend::AgentTicketCompose###ArticleFreeText

Description	Value
Description:	Shown article free text options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketCompose'}- >{'ArticleFreeText'} = { '1' => '0', '2' => '0', '3' => '0' };</pre>

Description	Value
	<code>};</code>

Frontend::Agent::Ticket::ViewCustomer

Ticket::Frontend::AgentTicketCustomer###Permission

Description	Value
Description:	Required permissions to use this option.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCustomer
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketCustomer'}->{'Permission'} = 'customer';</code>

Ticket::Frontend::AgentTicketCustomer###RequiredLock

Description	Value
Description:	A ticket lock is required. In case the ticket isn't locked, the tickets get locked and the current agent will be set automatically as ticket owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCustomer
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketCustomer'}->{'RequiredLock'} = '0';</code>

Frontend::Agent::Ticket::ViewEmailNew

Ticket::Frontend::AgentTicketEmail###Priority

Description	Value
Description:	Default priority for new email tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Priority'} = '3 normal';</code>

Ticket::Frontend::AgentTicketEmail###ArticleType

Description	Value
Description:	Default article type for new email tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'ArticleType'} = 'email-external';</pre>

Ticket::Frontend::AgentTicketEmail###SenderType

Description	Value
Description:	Default sender type for new email tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'SenderType'} = 'agent';</pre>

Ticket::Frontend::AgentTicketEmail###Subject

Description	Value
Description:	Default subject for new email tickets (e.g. 'email Outbound').
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Subject'} = '';</pre>

Ticket::Frontend::AgentTicketEmail###Body

Description	Value
Description:	Default text for new email tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1

Description	Value
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Body'} = '';</pre>

Ticket::Frontend::AgentTicketEmail###StateDefault

Description	Value
Description:	Default next state.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'StateDefault'} = 'open';</pre>

Ticket::Frontend::AgentTicketEmail###StateType

Description	Value
Description:	Next possible states after an email ticket.
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>

Ticket::Frontend::AgentTicketEmail###TicketFreeText

Description	Value
Description:	Shown ticket free text options. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	<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>

Ticket::Frontend::AgentTicketEmail###TicketFreeTime

Description	Value
Description:	Shown ticket free time options. 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'}- >{'TicketFreeTime'} = { '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' }; </pre>

Ticket::Frontend::AgentTicketEmail###ArticleFreeText

Description	Value
Description:	Shown article free text options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'ArticleFreeText'} = { '1' => '0', '2' => '0', '3' => '0' };</pre>

Ticket::Frontend::AgentTicketEmail###HistoryType

Description	Value
Description:	Standard history type for new email tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'HistoryType'} = 'EmailAgent';</pre>

Ticket::Frontend::AgentTicketEmail###HistoryComment

Description	Value
Description:	Default history comment for new email tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'HistoryComment'} = '';</pre>

Frontend::Agent::Ticket::ViewEscalation

Ticket::Frontend::AgentTicketEscalationView###ViewableTicketsPage

Description	Value
Description:	Shows all open tickets no matter wether locked or not by an agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	<pre>\$Self- >{'Ticket::Frontend::AgentTicketEscalationView'}- >{'ViewableTicketsPage'} = '50';</pre>

Ticket::Frontend::AgentTicketEscalationView###SortBy::Default

Description	Value
Description:	Queue sort by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketEscalationView'}- >{'SortBy::Default'} = 'EscalationTime';</pre>

Ticket::Frontend::AgentTicketEscalationView###Order::Default

Description	Value
Description:	Queue order default (after priority sort). ASC: oldest on top. DESC: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketEscalationView'}- >{'Order::Default'} = 'Up';</pre>

Frontend::Agent::Ticket::ViewForward

Ticket::Frontend::AgentTicketForward###Permission

Description	Value
Description:	Required permissions to use this option.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketForward'}- >{'Permission'} = 'forward';</pre>

Ticket::Frontend::AgentTicketForward###RequiredLock

Description	Value
Description:	A ticket lock is required. In case the ticket isn't locked, the tickets get locked and the current agent will be set automatically as ticket owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketForward'}->{'RequiredLock'} = '1';</pre>

Ticket::Frontend::AgentTicketForward###StateDefault

Description	Value
Description:	Default next state after forwarding a ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketForward'}->{'StateDefault'} = 'closed successful';</pre>

Ticket::Frontend::AgentTicketForward###StateType

Description	Value
Description:	Next possible states after forwarding a ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketForward'}->{'StateType'} = ['open', 'closed'];</pre>

Ticket::Frontend::AgentTicketForward###ArticleTypeDefault

Description	Value
Description:	Default type of forwarded message.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketForward'}->{'ArticleTypeDefault'} = 'email-external';</pre>

Ticket::Frontend::AgentTicketForward###ArticleTypes

Description	Value
Description:	Possible article types.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketForward'}->{'ArticleTypes'} = ['email-external', 'email-internal'];</pre>

Ticket::Frontend::AgentTicketForward###TicketFreeText

Description	Value
Description:	Shown ticket free text options. 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'}->{'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', }</pre>

Description	Value
	<pre>'6' => '0', '7' => '0', '8' => '0', '9' => '0' };</pre>

Ticket::Frontend::AgentTicketForward###TicketFreeTime

Description	Value
Description:	Shown ticket free time options. 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>

Frontend::Agent::Ticket::ViewFreeText

Ticket::Frontend::AgentTicketFreeText###Permission

Description	Value
Description:	Required permissions to use this option.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}- >{'Permission'} = 'rw';</pre>

Ticket::Frontend::AgentTicketFreeText###RequiredLock

Description	Value
Description:	A ticket lock is required. In case the ticket isn't locked, the ticket gets locked and the current agent will be set automatically as ticket owner.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'RequiredLock'} = '0';</pre>

Ticket::Frontend::AgentTicketFreeText###TicketType

Description	Value
Description:	If you want to set the ticket type (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'TicketType'} = '1';</pre>

Ticket::Frontend::AgentTicketFreeText###Service

Description	Value
Description:	If you want to set the service (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Service'} = '1';</pre>

Ticket::Frontend::AgentTicketFreeText###Owner

Description	Value
Description:	If you want to set the owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Owner'} = '0';</pre>

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:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'OwnerMandatory'} = '0';</pre>

Ticket::Frontend::AgentTicketFreeText###Responsible

Description	Value
Description:	If you want to set the responsible agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Responsible'} = '0';</pre>

Ticket::Frontend::AgentTicketFreeText###State

Description	Value
Description:	Would you like to set the state of a ticket if a note is added by an agent?
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'State'} = '0';</pre>

Ticket::Frontend::AgentTicketFreeText###StateType

Description	Value
Description:	Default next states after adding a note.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1

Description	Value
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}- >{'StateType'} = ['open', 'closed', 'pending reminder', 'pending auto'];</pre>

Ticket::Frontend::AgentTicketFreeText###StateDefault

Description	Value
Description:	Default next state.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}- >{'StateDefault'} = 'open';</pre>

Ticket::Frontend::AgentTicketFreeText###Note

Description	Value
Description:	Show note fields.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}- >{'Note'} = '0';</pre>

Ticket::Frontend::AgentTicketFreeText###Subject

Description	Value
Description:	Default note subject.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Subject'} = '\$Text{"Note"}';</code>

Ticket::Frontend::AgentTicketFreeText###Body

Description	Value
Description:	Default note text.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Body'} = '';</code>

Ticket::Frontend::AgentTicketFreeText###InvolvedAgent

Description	Value
Description:	Show selection of involved agents.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'InvolvedAgent'} = '0';</code>

Ticket::Frontend::AgentTicketFreeText###InformAgent

Description	Value
Description:	Show selection of agents to inform (all agents with note permissions on the queue/ticket).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'InformAgent'} = '0';</code>

Ticket::Frontend::AgentTicketFreeText###ArticleTypeDefault

Description	Value
Description:	Default note type.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ArticleTypeDefault'} = 'note-internal';</pre>

Ticket::Frontend::AgentTicketFreeText###ArticleTypes

Description	Value
Description:	Specify the different note types that you want to use in your system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ArticleTypes'} = { 'note-external' => '1', 'note-internal' => '1', 'note-report' => '0' };</pre>

Ticket::Frontend::AgentTicketFreeText###Priority

Description	Value
Description:	Show priority options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Priority'} = '0';</pre>

Ticket::Frontend::AgentTicketFreeText###PriorityDefault

Description	Value
Description:	Default priority options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText

Description	Value
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'PriorityDefault'} = '3 normal';</code>

Ticket::Frontend::AgentTicketFreeText###Title

Description	Value
Description:	Show title fields.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Title'} = '1';</code>

Ticket::Frontend::AgentTicketFreeText###TicketFreeText

Description	Value
Description:	Shown ticket free text options. 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>

Description	Value
	<code>};</code>

Ticket::Frontend::AgentTicketFreeText###TicketFreeTime

Description	Value
Description:	Shown ticket free time options. 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>

Ticket::Frontend::AgentTicketFreeText###ArticleFreeText

Description	Value
Description:	Shown article free text options.
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>

Ticket::Frontend::AgentTicketFreeText###HistoryType

Description	Value
Description:	History type for this action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText

Description	Value
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'HistoryType'} = 'AddNote';</code>

Ticket::Frontend::AgentTicketFreeText###HistoryComment

Description	Value
Description:	History comment for this action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'HistoryComment'} = '%FreeText';</code>

Frontend::Agent::Ticket::ViewHistory

Ticket::Frontend::HistoryOrder

Description	Value
Description:	Show history order reverse in the agent interface under ticket history.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewHistory
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::HistoryOrder'} = 'normal';</code>

Frontend::Agent::Ticket::ViewMailbox

Ticket::Frontend::AgentTicketLockedView###SortBy::Default

Description	Value
Description:	Mailbox sort by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketLockedView'}- >{'SortBy::Default'} = 'Age';</pre>

Ticket::Frontend::AgentTicketLockedView###Order::Default

Description	Value
Description:	Mailbox order default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketLockedView'}- >{'Order::Default'} = 'Up';</pre>

Ticket::Frontend::AgentTicketResponsibleView###SortBy::Default

Description	Value
Description:	Mailbox sort by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsibleView'}- >{'SortBy::Default'} = 'Age';</pre>

Ticket::Frontend::AgentTicketResponsibleView###Order::Default

Description	Value
Description:	Mailbox order default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsibleView'}- >{'Order::Default'} = 'Up';</pre>

Ticket::Frontend::AgentTicketWatchView###SortBy::Default

Description	Value
Description:	Mailbox sort by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketWatchView'}->{'SortBy::Default'} = 'Age';</code>

Ticket::Frontend::AgentTicketWatchView###Order::Default

Description	Value
Description:	Mailbox order default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketWatchView'}->{'Order::Default'} = 'Up';</code>

Frontend::Agent::Ticket::ViewMerge**Ticket::Frontend::AgentTicketMerge###Permission**

Description	Value
Description:	Required permissions to use this option.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'Permission'} = 'rw';</code>

Ticket::Frontend::AgentTicketMerge###RequiredLock

Description	Value
Description:	A ticket lock is required. In case the ticket isn't locked, the tickets get locked and the current agent will be set automatically as ticket owner.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'RequiredLock'} = '1';</pre>

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:	<pre>\$Self->{'Ticket::Frontend::MergeText'} = 'Your email with ticket number "<OTRS_TICKET>" is merged to "<OTRS_MERGE_TO_TICKET>".';</pre>

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:	<pre>\$Self->{'Ticket::Frontend::AutomaticMergeText'} = 'Merged Ticket <OTRS_TICKET> to <OTRS_MERGE_TO_TICKET>.';</pre>

Frontend::Agent::Ticket::ViewMove

Ticket::Frontend::MoveType

Description	Value
Description:	The queue selection can be shown in a dropdown list or in a new window. If "New Window" is set you can add a move note to the ticket.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::MoveType'} = 'form';</code>

Ticket::Frontend::AgentTicketMove###State

Description	Value
Description:	Set new ticket state in ticket move screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketMove'}->{'State'} = '1';</code>

Ticket::DefaultNextMoveStateType

Description	Value
Description:	Possible states after a ticket is moved.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::DefaultNextMoveStateType'} = ['open', 'closed'];</code>

Ticket::Frontend::AgentTicketMove###Priority

Description	Value
Description:	Show priority options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	<code>\$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Priority'} = '0';</code>

Ticket::Frontend::AgentTicketMove###TicketFreeText

Description	Value
Description:	Shown ticket free text options. 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'}->{'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>

Ticket::Frontend::AgentTicketMove###TicketFreeTime

Description	Value
Description:	Shown ticket free time options. 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', </pre>

Description	Value
	<pre>'3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

Frontend::Agent::Ticket::ViewNote

Ticket::Frontend::AgentTicketNote###Permission

Description	Value
Description:	Required permissions to use this option.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}- >{'Permission'} = 'note';</pre>

Ticket::Frontend::AgentTicketNote###RequiredLock

Description	Value
Description:	A ticket lock is required. In case the ticket isn't locked, the tickets get locked and the current agent will be set automatically as ticket owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}- >{'RequiredLock'} = '0';</pre>

Ticket::Frontend::AgentTicketNote###TicketType

Description	Value
Description:	If you want to set the ticket type (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}- >{'TicketType'} = '0';</pre>

Ticket::Frontend::AgentTicketNote###Service

Description	Value
Description:	If you want to set the service (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Service'} = '0';</pre>

Ticket::Frontend::AgentTicketNote###Owner

Description	Value
Description:	If you want to set the owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Owner'} = '0';</pre>

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:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'OwnerMandatory'} = '0';</pre>

Ticket::Frontend::AgentTicketNote###Responsible

Description	Value
Description:	If you want to set the responsible.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}- >{'Responsible'} = '0';</pre>

Ticket::Frontend::AgentTicketNote###State

Description	Value
Description:	Would you like to set the state of a ticket if a note is created by an agent?
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}- >{'State'} = '0';</pre>

Ticket::Frontend::AgentTicketNote###StateType

Description	Value
Description:	Default next states after adding a note.
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>

Ticket::Frontend::AgentTicketNote###StateDefault

Description	Value
Description:	Default next state.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}- >{'StateDefault'} = 'open';</pre>

Ticket::Frontend::AgentTicketNote###Note

Description	Value
Description:	Show note fields.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}- >{'Note'} = '1';</pre>

Ticket::Frontend::AgentTicketNote###Subject

Description	Value
Description:	Default note subject.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}- >{'Subject'} = '\$Text{"Note"}';</pre>

Ticket::Frontend::AgentTicketNote###Body

Description	Value
Description:	Default note text.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}- >{'Body'} = '';</pre>

Ticket::Frontend::AgentTicketNote###InvolvedAgent

Description	Value
Description:	Show selection of involved agents.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}- >{'InvolvedAgent'} = '0';</pre>

Ticket::Frontend::AgentTicketNote###InformAgent

Description	Value
Description:	Show selection of agents to inform (all agents with note permissions on the queue/ticket).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}- >{'InformAgent'} = '0';</pre>

Ticket::Frontend::AgentTicketNote###ArticleTypeDefault

Description	Value
Description:	Default note type.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}- >{'ArticleTypeDefault'} = 'note-internal';</pre>

Ticket::Frontend::AgentTicketNote###ArticleTypes

Description	Value
Description:	Specify the different note types, that you want to use in your system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}- >{'ArticleTypes'} = { 'note-external' => '1', 'note-internal' => '1', 'note-report' => '0'</pre>

Description	Value
	<code>};</code>

Ticket::Frontend::AgentTicketNote###Priority

Description	Value
Description:	Show priority options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Priority'} = '0';</code>

Ticket::Frontend::AgentTicketNote###PriorityDefault

Description	Value
Description:	Default priority options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'PriorityDefault'} = '3 normal';</code>

Ticket::Frontend::AgentTicketNote###Title

Description	Value
Description:	Show title fields.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Title'} = '0';</code>

Ticket::Frontend::AgentTicketNote###TicketFreeText

Description	Value
Description:	Shown ticket free text options. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket

Description	Value
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>

Ticket::Frontend::AgentTicketNote###TicketFreeTime

Description	Value
Description:	Shown ticket free time options. 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>

Ticket::Frontend::AgentTicketNote###ArticleFreeText

Description	Value
Description:	Shown article free text options.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}- >{'ArticleFreeText'} = { '1' => '0', '2' => '0', '3' => '0' };</pre>

Ticket::Frontend::AgentTicketNote###HistoryType

Description	Value
Description:	History type for this action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}- >{'HistoryType'} = 'AddNote';</pre>

Ticket::Frontend::AgentTicketNote###HistoryComment

Description	Value
Description:	History comment for this action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketNote'}- >{'HistoryComment'} = '%%Note';</pre>

Frontend::Agent::Ticket::ViewOwner

Ticket::Frontend::AgentTicketOwner###Permission

Description	Value
Description:	Required permissions to use this option.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Permission'} = 'owner';</pre>

Ticket::Frontend::AgentTicketOwner###RequiredLock

Description	Value
Description:	A ticket lock is required. In case the ticket isn't locked, the tickets get locked and the current agent will be set automatically as ticket owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'RequiredLock'} = '0';</pre>

Ticket::Frontend::AgentTicketOwner###TicketType

Description	Value
Description:	If you want to set the ticket type (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'TicketType'} = '0';</pre>

Ticket::Frontend::AgentTicketOwner###Service

Description	Value
Description:	If you want to set the service (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Service'} = '0';</pre>

Ticket::Frontend::AgentTicketOwner###Owner

Description	Value
Description:	If you want to set the owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Owner'} = '1';</pre>

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:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'OwnerMandatory'} = '1';</pre>

Ticket::Frontend::AgentTicketOwner###Responsible

Description	Value
Description:	If you want to set the responsible.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Responsible'} = '0';</pre>

Ticket::Frontend::AgentTicketOwner###State

Description	Value
Description:	Would you like to set the state of a ticket if a note is created by an agent?
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'State'} = '0';</pre>

Ticket::Frontend::AgentTicketOwner###StateType

Description	Value
Description:	Default next states after adding a note.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'StateType'} = ['open', 'pending reminder', 'pending auto'];</pre>

Ticket::Frontend::AgentTicketOwner###StateDefault

Description	Value
Description:	Default next state.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'StateDefault'} = 'open';</pre>

Ticket::Frontend::AgentTicketOwner###Note

Description	Value
Description:	Show note fields.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Note'} = '1';</pre>

Ticket::Frontend::AgentTicketOwner###Subject

Description	Value
Description:	Default note subject.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Subject'} = '\$Text{"Owner Update"}!';</pre>

Ticket::Frontend::AgentTicketOwner###Body

Description	Value
Description:	Default note text.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Body'} = '';</pre>

Ticket::Frontend::AgentTicketOwner###InvolvedAgent

Description	Value
Description:	Show selection of involved agents.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'InvolvedAgent'} = '0';</pre>

Ticket::Frontend::AgentTicketOwner###InformAgent

Description	Value
Description:	Show selection of agents to inform (all agents with note permissions on the queue/ticket).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner

Description	Value
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}- >{'InformAgent'} = '0';</pre>

Ticket::Frontend::AgentTicketOwner###ArticleTypeDefault

Description	Value
Description:	Default note type.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}- >{'ArticleTypeDefault'} = 'note-internal';</pre>

Ticket::Frontend::AgentTicketOwner###ArticleTypes

Description	Value
Description:	Specify the different note types, that you want to use in your system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}- >{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };</pre>

Ticket::Frontend::AgentTicketOwner###Priority

Description	Value
Description:	Show priority options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Priority'} = '0';</code>

Ticket::Frontend::AgentTicketOwner###PriorityDefault

Description	Value
Description:	Default priority options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'PriorityDefault'} = '3 normal';</code>

Ticket::Frontend::AgentTicketOwner###Title

Description	Value
Description:	Show title fields.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Title'} = '0';</code>

Ticket::Frontend::AgentTicketOwner###TicketFreeText

Description	Value
Description:	Shown ticket free text options. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'TicketFreeText'} = { '1' => '0', '10' => '0', '11' => '0', '12' => '0', '13' => '0', '14' => '0',</code>

Description	Value
	<pre>'15' => '0', '16' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0', '7' => '0', '8' => '0', '9' => '0' };</pre>

Ticket::Frontend::AgentTicketOwner###TicketFreeTime

Description	Value
Description:	Shown ticket free time options. 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>

Ticket::Frontend::AgentTicketOwner###ArticleFreeText

Description	Value
Description:	Shown article free text options.
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>

Ticket::Frontend::AgentTicketOwner###HistoryType

Description	Value
Description:	History type for this action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'HistoryType'} = 'AddNote';</pre>

Ticket::Frontend::AgentTicketOwner###HistoryComment

Description	Value
Description:	History comment for this action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'HistoryComment'} = '%%Owner';</pre>

Frontend::Agent::Ticket::ViewPending**Ticket::Frontend::AgentTicketPending###Permission**

Description	Value
Description:	Required permissions to use this option.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Permission'} = 'pending';</pre>

Ticket::Frontend::AgentTicketPending###RequiredLock

Description	Value
Description:	A ticket lock is required. In case the ticket isn't locked, the tickets get locked and the current agent will be set automatically as ticket owner.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'RequiredLock'} = '1';</code>

Ticket::Frontend::AgentTicketPending###TicketType

Description	Value
Description:	If you want to set the ticket type (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>

Ticket::Frontend::AgentTicketPending###Service

Description	Value
Description:	If you want to set the service (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>

Ticket::Frontend::AgentTicketPending###Owner

Description	Value
Description:	If you want to set the owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Owner'} = '0';</code>

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:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'OwnerMandatory'} = '0';</pre>

Ticket::Frontend::AgentTicketPending###Responsible

Description	Value
Description:	If you want to set the responsible.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Responsible'} = '0';</pre>

Ticket::Frontend::AgentTicketPending###State

Description	Value
Description:	Would you like to set the state of a ticket if a note is created by an agent?
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'State'} = '1';</pre>

Ticket::Frontend::AgentTicketPending###StateType

Description	Value
Description:	Default next states after adding a note.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending

Description	Value
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'StateType'} = ['pending reminder', 'pending auto'];</pre>

Ticket::Frontend::AgentTicketPending###StateDefault

Description	Value
Description:	Default next state.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'StateDefault'} = 'pending reminder';</pre>

Ticket::Frontend::AgentTicketPending###Note

Description	Value
Description:	Show note fields.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Note'} = '1';</pre>

Ticket::Frontend::AgentTicketPending###Subject

Description	Value
Description:	Default note subject.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Subject'} = '\$Text{"Pending"}!';</code>

Ticket::Frontend::AgentTicketPending###Body

Description	Value
Description:	Default note text.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Body'} = '';</code>

Ticket::Frontend::AgentTicketPending###InvolvedAgent

Description	Value
Description:	Show selection of involved agents.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'InvolvedAgent'} = '0';</code>

Ticket::Frontend::AgentTicketPending###InformAgent

Description	Value
Description:	Show selection of agents to inform (all agents with note permissions on the queue/ticket).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'InformAgent'} = '0';</code>

Ticket::Frontend::AgentTicketPending###ArticleTypeDefault

Description	Value
Description:	Default note type.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'ArticleTypeDefault'} = 'note-internal';</pre>

Ticket::Frontend::AgentTicketPending###ArticleTypes

Description	Value
Description:	Specify the different note types, that you want to use in your 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>

Ticket::Frontend::AgentTicketPending###Priority

Description	Value
Description:	Show priority options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Priority'} = '0';</pre>

Ticket::Frontend::AgentTicketPending###PriorityDefault

Description	Value
Description:	Default priority options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending

Description	Value
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'PriorityDefault'} = '3 normal';</code>

Ticket::Frontend::AgentTicketPending###Title

Description	Value
Description:	Show title fields.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Title'} = '0';</code>

Ticket::Frontend::AgentTicketPending###TicketFreeText

Description	Value
Description:	Shown ticket free text options. 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>

Description	Value
	<code>};</code>

Ticket::Frontend::AgentTicketPending###TicketFreeTime

Description	Value
Description:	Shown ticket free time options. 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'}->{'TicketFreeTime'} = { '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

Ticket::Frontend::AgentTicketPending###ArticleFreeText

Description	Value
Description:	Shown article free text options.
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>

Ticket::Frontend::AgentTicketPending###HistoryType

Description	Value
Description:	History type for this action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1

Description	Value
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'HistoryType'} = 'AddNote';</code>

Ticket::Frontend::AgentTicketPending###HistoryComment

Description	Value
Description:	History comment for this action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPending'}->{'HistoryComment'} = '%Pending';</code>

Frontend::Agent::Ticket::ViewPhoneNew

Ticket::Frontend::AgentTicketPhone###Priority

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

Ticket::Frontend::AgentTicketPhone###ArticleType

Description	Value
Description:	Default article type for new phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'ArticleType'} = 'phone';</code>

Ticket::Frontend::AgentTicketPhone###SenderType

Description	Value
Description:	Default sender type for new phone ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPhone'}- >{'SenderType'} = 'customer';</pre>

Ticket::Frontend::AgentTicketPhone###Subject

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

Ticket::Frontend::AgentTicketPhone###Body

Description	Value
Description:	Default note text for new telephone tickets. E.g 'New ticket via call'.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPhone'}- >{'Body'} = '';</pre>

Ticket::Frontend::AgentTicketPhone###StateDefault

Description	Value
Description:	Default next state for new phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPhone'}- >{'StateDefault'} = 'open';</pre>

Ticket::Frontend::AgentTicketPhone###StateType

Description	Value
Description:	Next possible states after a new phone ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPhone'}- >{'StateType'} = ['open', 'pending auto', 'pending reminder', 'closed'];</pre>

Ticket::Frontend::AgentTicketPhone###TicketFreeText

Description	Value
Description:	Shown ticket free text options. 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'}- >{'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', }</pre>

Description	Value
	<pre>'8' => '0', '9' => '0' };</pre>

Ticket::Frontend::AgentTicketPhone###TicketFreeTime

Description	Value
Description:	Shown ticket free time options. 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>

Ticket::Frontend::AgentTicketPhone###ArticleFreeText

Description	Value
Description:	Shown article free text options.
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>

Ticket::Frontend::AgentTicketPhone###HistoryType

Description	Value
Description:	Default history type for new phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew

Description	Value
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'HistoryType'} = 'PhoneCallCustomer';</pre>

Ticket::Frontend::AgentTicketPhone###HistoryComment

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

Ticket::Frontend::AgentTicketPhone###SplitLinkType

Description	Value
Description:	Default link type of splitted tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'SplitLinkType'} = { 'Direction' => 'Target', 'LinkType' => 'ParentChild' };</pre>

Frontend::Agent::Ticket::ViewPhoneOutbound

Ticket::Frontend::AgentTicketPhoneOutbound###Permission

Description	Value
Description:	Required permissions to use this option.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketPhoneOutbound'}- >{'Permission'} = 'phone';</pre>

Ticket::Frontend::AgentTicketPhoneOutbound###RequiredLock

Description	Value
Description:	A ticket lock is required. In case the ticket isn't locked, the tickets get locked and the current agent will be set automatically as ticket owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketPhoneOutbound'}- >{'RequiredLock'} = '1';</pre>

Ticket::Frontend::AgentTicketPhoneOutbound###ArticleType

Description	Value
Description:	Default note type for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketPhoneOutbound'}- >{'ArticleType'} = 'phone';</pre>

Ticket::Frontend::AgentTicketPhoneOutbound###SenderType

Description	Value
Description:	Default sender type for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketPhoneOutbound'}- >{'SenderType'} = 'agent';</pre>

Ticket::Frontend::AgentTicketPhoneOutbound###Subject

Description	Value
Description:	Default subject for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketPhoneOutbound'}- >{'Subject'} = '\$Text{"Phone call"}!';</pre>

Ticket::Frontend::AgentTicketPhoneOutbound###Body

Description	Value
Description:	Default note text for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketPhoneOutbound'}- >{'Body'} = '';</pre>

Ticket::Frontend::AgentTicketPhoneOutbound###State

Description	Value
Description:	Default next state after adding a phone note.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketPhoneOutbound'}- >{'State'} = 'closed successful';</pre>

Ticket::Frontend::AgentTicketPhoneOutbound###StateType

Description	Value
Description:	Next possible states after phone note.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound

Description	Value
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self- >{'Ticket::Frontend::AgentTicketPhoneOutbound'}- >{'StateType'} = ['open', 'pending auto', 'pending reminder', 'closed']; </pre>

Ticket::Frontend::AgentTicketPhoneOutbound###TicketFreeText

Description	Value
Description:	Shown ticket free text options. 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'}- >{'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>

Ticket::Frontend::AgentTicketPhoneOutbound###TicketFreeTime

Description	Value
Description:	Shown ticket free time options. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Description	Value
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>

Ticket::Frontend::AgentTicketPhoneOutbound###ArticleFreeText

Description	Value
Description:	Shown article free text options.
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>

Ticket::Frontend::AgentTicketPhoneOutbound###HistoryType

Description	Value
Description:	Default history type for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self- >{'Ticket::Frontend::AgentTicketPhoneOutbound'}- >{'HistoryType'} = 'PhoneCallAgent'; </pre>

Ticket::Frontend::AgentTicketPhoneOutbound###HistoryComment

Description	Value
Description:	Default history comment for phone tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'HistoryComment'} = '';</pre>

Frontend::Agent::Ticket::ViewPriority

Ticket::Frontend::AgentTicketPriority###Permission

Description	Value
Description:	Required permissions to use this option.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Permission'} = 'priority';</pre>

Ticket::Frontend::AgentTicketPriority###RequiredLock

Description	Value
Description:	A ticket lock is required. In case the ticket isn't locked, the tickets get locked and the current agent will be set automatically as ticket owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'RequiredLock'} = '1';</pre>

Ticket::Frontend::AgentTicketPriority###TicketType

Description	Value
Description:	If you want to set the ticket type (Ticket::Type needs to be activated).

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'TicketType'} = '0';</pre>

Ticket::Frontend::AgentTicketPriority###Service

Description	Value
Description:	If you want to set the service (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Service'} = '0';</pre>

Ticket::Frontend::AgentTicketPriority###Owner

Description	Value
Description:	If you want to set the owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Owner'} = '0';</pre>

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:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'OwnerMandatory'} = '0';</pre>

Ticket::Frontend::AgentTicketPriority###Responsible

Description	Value
Description:	If you want to set the responsible.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Responsible'} = '0';</pre>

Ticket::Frontend::AgentTicketPriority###State

Description	Value
Description:	Would you like to set the state of a ticket if a note is created by an agent?
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'State'} = '0';</pre>

Ticket::Frontend::AgentTicketPriority###StateType

Description	Value
Description:	Default next states after adding a note.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'StateType'} = ['open', 'pending reminder', 'pending auto'];</pre>

Ticket::Frontend::AgentTicketPriority###StateDefault

Description	Value
Description:	Default next state.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'StateDefault'} = 'open';</pre>

Ticket::Frontend::AgentTicketPriority###Note

Description	Value
Description:	Show note fields.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Note'} = '1';</pre>

Ticket::Frontend::AgentTicketPriority###Subject

Description	Value
Description:	Default note subject.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Subject'} = '\$Text{"Priority Update"}!';</pre>

Ticket::Frontend::AgentTicketPriority###Body

Description	Value
Description:	Default note text.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Body'} = '';</pre>

Ticket::Frontend::AgentTicketPriority###InvolvedAgent

Description	Value
Description:	Show selection of involved agents.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'InvolvedAgent'} = '0';</pre>

Ticket::Frontend::AgentTicketPriority###InformAgent

Description	Value
Description:	Show selection of agents to inform (all agents with note permissions on the queue/ticket).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'InformAgent'} = '0';</pre>

Ticket::Frontend::AgentTicketPriority###ArticleTypeDefault

Description	Value
Description:	Default note type.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ArticleTypeDefault'} = 'note-internal';</pre>

Ticket::Frontend::AgentTicketPriority###ArticleTypes

Description	Value
Description:	Specify the different note types, that you want to use in your system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority

Description	Value
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };</pre>

Ticket::Frontend::AgentTicketPriority###Priority

Description	Value
Description:	Show priority options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Priority'} = '1';</pre>

Ticket::Frontend::AgentTicketPriority###PriorityDefault

Description	Value
Description:	Default priority options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'PriorityDefault'} = '3 normal';</pre>

Ticket::Frontend::AgentTicketPriority###Title

Description	Value
Description:	Show title fields.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	<code>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Title'} = '0';</code>

Ticket::Frontend::AgentTicketPriority###TicketFreeText

Description	Value
Description:	Shown ticket free text options. 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'}->{'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>

Ticket::Frontend::AgentTicketPriority###TicketFreeTime

Description	Value
Description:	Shown ticket free time options. 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'}->{'TicketFreeTime'} = { '1' => '0', '2' => '0', </pre>

Description	Value
	<pre>'3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

Ticket::Frontend::AgentTicketPriority###ArticleFreeText

Description	Value
Description:	Shown article free text options.
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>

Ticket::Frontend::AgentTicketPriority###HistoryType

Description	Value
Description:	History type for this action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'HistoryType'} = 'AddNote';</pre>

Ticket::Frontend::AgentTicketPriority###HistoryComment

Description	Value
Description:	History comment for this action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'HistoryComment'} = '%%Priority';</pre>

Frontend::Agent::Ticket::ViewQueue

Ticket::Frontend::AgentTicketQueue###StripEmptyLines

Description	Value
Description:	Strip empty lines on ticket preview of queue view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'StripEmptyLines'} = '0';</pre>

Ticket::Frontend::AgentTicketQueue###ViewAllPossibleTickets

Description	Value
Description:	Show all ro and rw queues in the QueueView (not only rw queues)?
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'ViewAllPossibleTickets'} = '0';</pre>

Ticket::Frontend::AgentTicketQueue###HighlightAge1

Description	Value
Description:	Set the age (first level) for highlighting of queues containing untouched tickets older than x minutes.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'HighlightAge1'} = '1440';</pre>

Ticket::Frontend::AgentTicketQueue###HighlightColor1

Description	Value
Description:	Set the color for highlighting queues containing untouched tickets (first level).

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

Ticket::Frontend::AgentTicketQueue###HighlightAge2

Description	Value
Description:	Set the age (second level) highlighting of queues containing untouched tickets older than x minutes.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'HighlightAge2'} = '2880';</pre>

Ticket::Frontend::AgentTicketQueue###HighlightColor2

Description	Value
Description:	Set the color for highlighting queues containing untouched tickets (second level).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'HighlightColor2'} = 'red';</pre>

Ticket::Frontend::AgentTicketQueue###Blink

Description	Value
Description:	Activate the blinking mechanism of the queue with the oldest ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'Blink'} = '1';</pre>

Ticket::Frontend::AgentTicketQueue###QueueSort

Description	Value
Description:	Sort the tickets when a single queue is selected in the QueueView in ascending or descending order (after the tickets are sorted by priority). Values: 0 = ascending (oldest on top, default), 1 = descending (youngest on top). Use the QueueID for the key and 0 or 1 for value.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'QueueSort'} = { '3' => '0', '7' => '1' };</pre>

Ticket::Frontend::AgentTicketQueue###SortBy::Default

Description	Value
Description:	Default sort criteria for all queues displayed in the QueueView after sort by priority is done.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'SortBy::Default'} = 'Age';</pre>

Ticket::Frontend::AgentTicketQueue###Order::Default

Description	Value
Description:	Sort order for all queues in the QueueView after priority sort.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	<code>\$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'Order::Default'} = 'Up';</code>

Frontend::Agent::Ticket::ViewResponsible

Ticket::Frontend::AgentTicketResponsible###Permission

Description	Value
Description:	Required permissions to use this option.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Permission'} = 'responsible';</code>

Ticket::Frontend::AgentTicketResponsible###RequiredLock

Description	Value
Description:	A ticket lock is required. In case the ticket isn't locked, the ticket gets locked and the current agent will be set automatically as ticket owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'RequiredLock'} = '0';</code>

Ticket::Frontend::AgentTicketResponsible###TicketType

Description	Value
Description:	If you want to set the ticket type (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'TicketType'} = '0';</code>

Ticket::Frontend::AgentTicketResponsible###Service

Description	Value
Description:	If you want to set the service (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'Service'} = '0';</pre>

Ticket::Frontend::AgentTicketResponsible###Owner

Description	Value
Description:	If you want to set the owner.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'Owner'} = '0';</pre>

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
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'OwnerMandatory'} = '0';</pre>

Ticket::Frontend::AgentTicketResponsible###Responsible

Description	Value
Description:	If you want to set the responsible.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible

Description	Value
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'Responsible'} = '1';</pre>

Ticket::Frontend::AgentTicketResponsible###State

Description	Value
Description:	Would you like to set the state of a ticket if a note is created by an agent?
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'State'} = '0';</pre>

Ticket::Frontend::AgentTicketResponsible###StateType

Description	Value
Description:	Default next states after adding a note.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'StateType'} = ['open', 'pending reminder', 'pending auto'];</pre>

Ticket::Frontend::AgentTicketResponsible###StateDefault

Description	Value
Description:	Default next state.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1

Description	Value
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'StateDefault'} = 'open';</pre>

Ticket::Frontend::AgentTicketResponsible###Note

Description	Value
Description:	Show note fields.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'Note'} = '1';</pre>

Ticket::Frontend::AgentTicketResponsible###Subject

Description	Value
Description:	Default note subject.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'Subject'} = '\$Text{"Responsible Update"}!';</pre>

Ticket::Frontend::AgentTicketResponsible###Body

Description	Value
Description:	Default note text.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'Body'} = '';</pre>

Ticket::Frontend::AgentTicketResponsible###InvolvedAgent

Description	Value
Description:	Show selection of involved agents.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'InvolvedAgent'} = '0';</pre>

Ticket::Frontend::AgentTicketResponsible###InformAgent

Description	Value
Description:	Show selection of agents to inform (all agents with note permissions on the queue/ticket).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'InformAgent'} = '0';</pre>

Ticket::Frontend::AgentTicketResponsible###ArticleTypeDefault

Description	Value
Description:	Default note type.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'ArticleTypeDefault'} = 'note-internal';</pre>

Ticket::Frontend::AgentTicketResponsible###ArticleTypes

Description	Value
Description:	Specify the different note types, that you want to use in your system.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };</pre>

Ticket::Frontend::AgentTicketResponsible###Priority

Description	Value
Description:	Show priority options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'Priority'} = '0';</pre>

Ticket::Frontend::AgentTicketResponsible###PriorityDefault

Description	Value
Description:	Default priority options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'PriorityDefault'} = '3 normal';</pre>

Ticket::Frontend::AgentTicketResponsible###Title

Description	Value
Description:	Show title fields.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible

Description	Value
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'Title'} = '1';</pre>

Ticket::Frontend::AgentTicketResponsible###TicketFreeText

Description	Value
Description:	Shown ticket free text options. 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>

Ticket::Frontend::AgentTicketResponsible###TicketFreeTime

Description	Value
Description:	Shown ticket free time options. 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:	<pre> \$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'TicketFreeTime'} = { '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' }; </pre>

Ticket::Frontend::AgentTicketResponsible###ArticleFreeText

Description	Value
Description:	Shown article free text options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'ArticleFreeText'} = { '1' => '0', '2' => '0', '3' => '0' }; </pre>

Ticket::Frontend::AgentTicketResponsible###HistoryType

Description	Value
Description:	History type for this action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self- >{'Ticket::Frontend::AgentTicketResponsible'}- >{'HistoryType'} = 'AddNote'; </pre>

Ticket::Frontend::AgentTicketResponsible###HistoryComment

Description	Value
Description:	History comment for this action.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketResponsible'}- >{'HistoryComment'} = '%Responsible';</pre>

Frontend::Agent::Ticket::ViewSearch

Ticket::Frontend::AgentTicketSearch###ExtendedSearchCondition

Description	Value
Description:	Extended search condition in ticket search. 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:	<pre>\$Self->{'Ticket::Frontend::AgentTicketSearch'}- >{'ExtendedSearchCondition'} = '1';</pre>

Ticket::Frontend::AgentTicketSearch###SearchLimit

Description	Value
Description:	Search limit for ticket frontend.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketSearch'}- >{'SearchLimit'} = '2000';</pre>

Ticket::Frontend::AgentTicketSearch###SearchPageShown

Description	Value
Description:	Default of shown results per page.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchPageShown'} = '40';</pre>

Ticket::Frontend::AgentTicketSearch###SearchViewableTicketLines

Description	Value
Description:	Lines of each ticket that are shown by the search utility.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchViewableTicketLines'} = '10';</pre>

Ticket::Frontend::AgentTicketSearch###TicketFreeText

Description	Value
Description:	Shown ticket free text options.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	<pre>\$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',</pre>

Description	Value
	<code>'9' => '0'</code> <code>};</code>

Ticket::Frontend::AgentTicketSearch###TicketFreeTime

Description	Value
Description:	Shown ticket free time options. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'TicketFreeTime'} = { '1' => '0', '2' => '0', '3' => '0', '4' => '0', '5' => '0', '6' => '0' };</pre>

Ticket::Frontend::AgentTicketSearch###SortBy::Default

Description	Value
Description:	Search result sort by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SortBy::Default'} = 'Age';</pre>

Ticket::Frontend::AgentTicketSearch###Order::Default

Description	Value
Description:	Search result order default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Order::Default'} = 'Down';</code>

Ticket::Frontend::AgentTicketSearch###SearchArticleCSVTree

Description	Value
Description:	Export whole article tree in search result export (take care of the system performance!).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchArticleCSVTree'} = '0';</code>

Ticket::Frontend::AgentTicketSearch###SearchCSVData

Description	Value
Description:	Used data to export the search result in CSV format..
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
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',</code>

Description	Value
	<pre>'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'];</pre>

Ticket::Frontend::AgentTicketSearch###ArticleCreateTime

Description	Value
Description:	Include article create times in search.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::AgentTicketSearch'}- >{'ArticleCreateTime'} = '0';</pre>

Frontend::Agent::Ticket::ViewStatus

Ticket::Frontend::AgentTicketStatusView###ViewableTicketsPage

Description	Value
Description:	Shows all open tickets no matter wether locked or not by an agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus

Description	Value
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketStatusView'}- >{'ViewableTicketsPage'} = '50';</pre>

Ticket::Frontend::AgentTicketStatusView###SortBy::Default

Description	Value
Description:	Queue sort by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketStatusView'}- >{'SortBy::Default'} = 'Age';</pre>

Ticket::Frontend::AgentTicketStatusView###Order::Default

Description	Value
Description:	Queue order default (after priority sort). ASC: oldest on top. DESC: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::AgentTicketStatusView'}- >{'Order::Default'} = 'Down';</pre>

Frontend::Agent::Ticket::ViewZoom

Ticket::Frontend::PlainView

Description	Value
Description:	Show also in ZoomView the plain link for emails in article tree.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1

Description	Value
Config-Setting:	<code>\$Self->{'Ticket::Frontend::PlainView'} = '0';</code>

Ticket::Frontend::ZoomExpand

Description	Value
Description:	Show all articles to a ticket in the ZoomView without selecting which article to show?
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::ZoomExpand'} = '0';</code>

Ticket::Frontend::ZoomExpandSort

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

Ticket::ZoomAttachmentDisplay

Description	Value
Description:	Shows an icon in the ticket zoom, if an article has an attachment.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::ZoomAttachmentDisplay'} = '1';</code>

Ticket::ZoomAttachmentDisplayCount

Description	Value
Description:	Shows a count of icons in the ticket zoom, if an article has an attachment.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::ZoomAttachmentDisplayCount'} = '3';</code>

Ticket::ZoomTimeDisplay

Description	Value
Description:	Displays the accounted time for an article in the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	0
Required:	0
Config-Setting:	<code>\$Self->{'Ticket::ZoomTimeDisplay'} = '0';</code>

Ticket::Frontend::TicketArticleFilter

Description	Value
Description:	Activates the article filter in ZoomView.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::TicketArticleFilter'} = '0';</code>

Ticket::Frontend::ZoomRichTextForce

Description	Value
Description:	Show article also on richt text even rich text is disabled.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::ZoomRichTextForce'} = '0';</code>

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:	<pre>\$Self->{'Ticket::Frontend::HTMLArticleHeightDefault'} = '100';</pre>

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
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::HTMLArticleHeightMax'} = '2500';</pre>

Frontend::Agent::TicketOverview

Ticket::Frontend::Overview###Small

Description	Value
Description:	Module registration for ticket small overview (CustomerInfo => 1 - shows also Customer-Info).
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::Overview'}->{'Small'} = { 'CustomerInfo' => '1', 'Image' => 'overviewsmall.png', 'ImageSelected' => 'overviewsmall-selected.png',</pre>

Description	Value
	<pre>'Module' => 'Kernel::Output::HTML::TicketOverviewSmall', 'Name' => 'Small', 'PageShown' => '25' };</pre>

Ticket::Frontend::OverviewSmall###ColumnHeader

Description	Value
Description:	Show in the small view the subject of the last customer article or the ticket title.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::OverviewSmall'}- >{'ColumnHeader'} = 'LastCustomerSubject';</pre>

Ticket::Frontend::Overview###Medium

Description	Value
Description:	Module registration for ticket medium overview (CustomerInfo => 1 - shows also Customer-Info).
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::Overview'}->{'Medium'} = { 'CustomerInfo' => '0', 'Image' => 'overviewmedium.png', 'ImageSelected' => 'overviewmedium-selected.png', 'Module' => 'Kernel::Output::HTML::TicketOverviewMedium', 'Name' => 'Medium', 'PageShown' => '20' };</pre>

Ticket::Frontend::Overview###Preview

Description	Value
Description:	Module registration for ticket preview overview (CustomerInfo => 1 - shows also Customer-Info, CustomerInfoMaxSize max. size (in characters) of Customer-Info).

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::Overview'}->{'Preview'} = { 'CustomerInfo' => '0', 'CustomerInfoMaxSize' => '18', 'Image' => 'overviewpreview.png', 'ImageSelected' => 'overviewpreview- selected.png', 'Module' => 'Kernel::Output::HTML::TicketOverviewPreview', 'Name' => 'Preview', 'PageShown' => '15' }; </pre>

Frontend::Customer

CustomerFrontend::CommonObject###QueueObject

Description	Value
Description:	Path to the file where all settings for the QueueObject object for the customer interface are stored.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'CustomerFrontend::CommonObject'}- >{'QueueObject'} = 'Kernel::System::Queue'; </pre>

CustomerFrontend::CommonObject###TicketObject

Description	Value
Description:	Path to the file where all settings for the TicketObject object for the customer interface are stored.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'CustomerFrontend::CommonObject'}- >{'TicketObject'} = 'Kernel::System::Ticket'; </pre>

CustomerFrontend::CommonParam###Action

Description	Value
Description:	Specify the common action for the Action parameter in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerFrontend::CommonParam'}- >{'Action'} = 'CustomerTicketOverView';</pre>

CustomerFrontend::CommonParam###TicketID

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

Frontend::Customer::ModuleRegistration

CustomerFrontend::Module###CustomerTicketOverView

Description	Value
Description:	Frontend module registration for the CustomerTicketOverview object in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}- >{'CustomerTicketOverView'} = { 'Description' => 'Overview of customer tickets', 'NavBar' => [{ 'AccessKey' => 'm', 'Block' => '', 'Description' => 'MyTickets', 'Image' => 'ticket.png',</pre>

Description	Value
	<pre> 'Link' => 'Action=CustomerTicketOverView&Type=MyTickets', 'Name' => 'MyTickets', 'NavBar' => '', 'Prio' => '110', 'Type' => '' }, { 'AccessKey' => 'c', 'Block' => '', 'Description' => 'CompanyTickets', 'Image' => 'tickets.png', 'Link' => 'Action=CustomerTicketOverView&Type=CompanyTickets', 'Name' => 'CompanyTickets', 'NavBar' => '', 'Prio' => '120', 'Type' => '' }], 'NavBarName' => 'Ticket', 'Title' => 'Overview' }; </pre>

CustomerFrontend::Module###CustomerTicketMessage

Description	Value
Description:	Frontend module registration for the CustomerTicketMessage object in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerFrontend::Module'}- >{'CustomerTicketMessage'} = { 'Description' => 'Create tickets', 'NavBar' => [{ 'AccessKey' => 'n', 'Block' => '', 'Description' => 'Create new Ticket', 'Image' => 'new.png', 'Link' => 'Action=CustomerTicketMessage', 'Name' => 'New Ticket', 'NavBar' => '', 'Prio' => '100', 'Type' => '' }], </pre>

Description	Value
	<pre>'NavBarName' => 'Ticket', 'Title' => 'New Ticket' };</pre>

CustomerFrontend::Module###CustomerTicketZoom

Description	Value
Description:	Frontend module registration for the CustomerTicketZoom object in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}- >{'CustomerTicketZoom'} = { 'Description' => 'Ticket zoom view', 'NavBarName' => 'Ticket', 'Title' => 'Zoom' };</pre>

CustomerFrontend::Module###CustomerTicketPrint

Description	Value
Description:	Customer Frontend module registration for the Customer Ticket Print Module.
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>

CustomerFrontend::Module###CustomerZoom

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

Description	Value
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}- >{'CustomerZoom'} = { 'Description' => 'compat mod', 'NavBarName' => '', 'Title' => '' };</pre>

CustomerFrontend::Module###CustomerTicketAttachment

Description	Value
Description:	Frontend module registration for the CustomerTicketAttachment object in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}- >{'CustomerTicketAttachment'} = { 'Description' => 'To download attachments', 'NavBarName' => '', 'Title' => '' };</pre>

CustomerFrontend::Module###CustomerTicketSearch

Description	Value
Description:	Frontend module registration for the CustomerTicketSearch object in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerFrontend::Module'}- >{'CustomerTicketSearch'} = { 'Description' => 'Customer ticket search', 'NavBar' => [{ 'AccessKey' => 's', 'Block' => '', 'Description' => 'Search', 'Image' => 'search.png', 'Link' => 'Action=CustomerTicketSearch', }] };</pre>

Description	Value
	<pre> 'Name' => 'Search', 'NavBar' => '', 'Prio' => '300', 'Type' => '' }], 'NavBarName' => 'Ticket', 'Title' => 'Search' }; </pre>

Frontend::Customer::Preferences

CustomerPreferencesGroups###ClosedTickets

Description	Value
Description:	All parameters for the ClosedTickets object in the customer preferences.
Group:	Ticket
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerPreferencesGroups'}- >{'ClosedTickets'} = { 'Activ' => '1', 'Colum' => 'Other Options', 'Data' => { '0' => 'No', '1' => 'Yes' }, 'DataSelected' => '1', 'Desc' => 'Show closed tickets.', 'Label' => 'Closed Tickets', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserShowClosedTickets', 'Prio' => '2000' }; </pre>

CustomerPreferencesGroups###ShownTickets

Description	Value
Description:	All parameters for the ShownTickets object in the customer preferences.
Group:	Ticket
SubGroup:	Frontend::Customer::Preferences
Valid:	1

Description	Value
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerPreferencesGroups'}- >{'ShownTickets'} = { 'Activ' => '1', 'Colum' => 'Frontend', 'Data' => { '15' => '15', '20' => '20', '25' => '25', '30' => '30' }, 'DataSelected' => '25', 'Desc' => 'Max. shown Tickets a page in Overview.', 'Label' => 'Shown Tickets', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserShowTickets', 'Prio' => '4000' }; </pre>

CustomerPreferencesGroups###RefreshTime

Description	Value
Description:	All parameters for the RefreshTime object in the customer preferences.
Group:	Ticket
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	<pre> \$Self->{'CustomerPreferencesGroups'}- >{'RefreshTime'} = { 'Activ' => '1', 'Colum' => 'Frontend', 'Data' => { '' => 'off', '10' => '10 minutes', '15' => '15 minutes', '2' => ' 2 minutes', '5' => ' 5 minutes', '7' => ' 7 minutes' }, 'DataSelected' => '', 'Desc' => 'Select your QueueView refresh time.', 'Label' => 'QueueView refresh time', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserRefreshTime', }; </pre>

Description	Value
	<code>'Prio' => '4000'</code> <code>};</code>

Frontend::Customer::Ticket::ViewNew

Ticket::Frontend::CustomerTicketMessage###NextScreenAfterNewTicket

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

Ticket::Frontend::CustomerTicketMessage###Priority

Description	Value
Description:	Allow customers to set the ticket priority.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<code>\$Self-</code> <code>>{'Ticket::Frontend::CustomerTicketMessage'}-</code> <code>>{'Priority'} = '1';</code>

Ticket::Frontend::CustomerTicketMessage###PriorityDefault

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

Ticket::Frontend::CustomerTicketMessage###StateDefault

Description	Value
Description:	Default state of new customer tickets.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::CustomerTicketMessage'}- >{'StateDefault'} = 'new';</pre>

Ticket::Frontend::CustomerTicketMessage###ArticleType

Description	Value
Description:	Default article type.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::CustomerTicketMessage'}- >{'ArticleType'} = 'webrequest';</pre>

Ticket::Frontend::CustomerTicketMessage###SenderType

Description	Value
Description:	Sender type for new tickets from the customer panel.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::Frontend::CustomerTicketMessage'}- >{'SenderType'} = 'customer';</pre>

Ticket::Frontend::CustomerTicketMessage###HistoryType

Description	Value
Description:	Default history type.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}- >{'HistoryType'} = 'WebRequestCustomer';</pre>

Ticket::Frontend::CustomerTicketMessage###HistoryComment

Description	Value
Description:	Comment for new history entries in the customer panel.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::CustomerTicketMessage'}- >{'HistoryComment'} = '';</pre>

CustomerPanelSelectionType

Description	Value
Description:	To: selection type. "Queue" shows all queues, "SystemAddress" displays all system addresses.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'CustomerPanelSelectionType'} = 'Queue';</pre>

CustomerPanelSelectionString

Description	Value
Description:	If "Queue" was selected for CustomerPanelSelectionType, you can specify a selection string (e. g. 'Queue: <Queue> - <QueueComment>').
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	0

Description	Value
Config-Setting:	<pre>\$Self->{'CustomerPanelSelectionString'} = '<Queue>';</pre>

CustomerPanelOwnSelection

Description	Value
Description:	Here you can insert the Queues, which the customer can select in the Customer-Interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	0
Required:	0
Config-Setting:	<pre>\$Self->{'CustomerPanelOwnSelection'} = { 'Junk' => 'First Queue', 'Misc' => 'Second Queue' };</pre>

CustomerPanel::NewTicketQueueSelectionModule

Description	Value
Description:	Module for To-selection in new ticket screen in the customer panel.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self-> >{'CustomerPanel::NewTicketQueueSelectionModule'} = 'Kernel::Output::HTML::CustomerNewTicketQueueSelectionGeneric';</pre>

Ticket::Frontend::CustomerTicketMessage###TicketFreeText

Description	Value
Description:	Shown ticket free text options. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	<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>

Ticket::Frontend::CustomerTicketMessage###TicketFreeTime

Description	Value
Description:	Shown ticket free time options. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
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>

Frontend::Customer::Ticket::ViewSearch

Ticket::CustomerTicketSearch::SearchLimit

Description	Value
Description:	Default limit for ticket search.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::CustomerTicketSearch::SearchLimit'} = '5000';</pre>

Ticket::CustomerTicketSearch::SearchPageShown

Description	Value
Description:	Default number of articles per page after searching through tickets.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self- >{'Ticket::CustomerTicketSearch::SearchPageShown'} = '40';</pre>

Ticket::CustomerTicketSearch::SortBy::Default

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

Ticket::CustomerTicketSearch::Order::Default

Description	Value
Description:	Search result order default.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1

Description	Value
Config-Setting:	<pre>\$Self->{'Ticket::CustomerTicketSearch::Order::Default'} = 'Down';</pre>

Ticket::Frontend::CustomerTicketSearch###ExtendedSearchCondition

Description	Value
Description:	Extended search condition in ticket search. 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:	<pre>\$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'ExtendedSearchCondition'} = '1';</pre>

Ticket::Frontend::CustomerTicketSearch###TicketFreeText

Description	Value
Description:	Shown ticket free text options.
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>

Ticket::Frontend::CustomerTicketSearch###TicketFreeTime

Description	Value
Description:	Shown ticket free time options. Possible settings: 0 = Disabled, 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>

Frontend::Customer::Ticket::ViewZoom

Ticket::Frontend::CustomerTicketZoom###NextScreenAfterFollowUp

Description	Value
Description:	Next screen after customer follow up ticket.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<pre> \$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'NextScreenAfterFollowUp'} = 'CustomerTicketOverView'; </pre>

Ticket::Frontend::CustomerTicketZoom###ArticleType

Description	Value
Description:	Type of note.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	<code>\$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'ArticleType'} = 'webrequest';</code>

Ticket::Frontend::CustomerTicketZoom###SenderType

Description	Value
Description:	Sender type for requests from the customer panel.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'SenderType'} = 'customer';</code>

Ticket::Frontend::CustomerTicketZoom###HistoryType

Description	Value
Description:	Default history type in the customer panel.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<code>\$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'HistoryType'} = 'FollowUp';</code>

Ticket::Frontend::CustomerTicketZoom###HistoryComment

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

Ticket::Frontend::CustomerTicketZoom###Priority

Description	Value
Description:	Allow customers to change the ticket priority.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'Priority'} = '1';</pre>

Ticket::Frontend::CustomerTicketZoom###PriorityDefault

Description	Value
Description:	Default priority of follow up customer tickets.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'PriorityDefault'} = '3 normal';</pre>

Ticket::Frontend::CustomerTicketZoom###State

Description	Value
Description:	Next compose state for customer tickets.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	<pre>\$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'State'} = '1';</pre>

Ticket::Frontend::CustomerTicketZoom###StateDefault

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

Ticket::Frontend::CustomerTicketZoom###StateType

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

Frontend::Queue::Preferences

QueuePreferences###Comment2

Description	Value
Description:	Parameters for the example comment 2 of queue attributes.
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>

Frontend::SLA::Preferences

SLAPreferences###Comment2

Description	Value
Description:	Parameters for the example comment 2 of sla attributes.
Group:	Ticket
SubGroup:	Frontend::SLA::Preferences

Description	Value
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>

Frontend::Service::Preferences

ServicePreferences###Comment2

Description	Value
Description:	Parameters for the example comment 2 of service attributes.
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.

PREAMBLE

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

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

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

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.

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

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

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.

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.

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."

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.

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.

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.

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.

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.