Documentation

OTRS 3.3 - Admin Manual

Build Date:

2013-04-30

OTRS 3.3 - Admin Manual

Авторские права © 2003-2013 OTRS AG

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

Эта работа защищена авторским правом OTRS AG.

Это руководство может быть скопировано целиком или частично, при условии, что это уведомление об авторских правах будет включено в каждую копию.

The source code of this document can be found at github, in the repository doc-admin.

UNIX является зарегистрированной торговой маркой компании X/Open Company Limited. Linux является зарегистрированной торговой маркой Линуса Торвальдса (Linus Torvalds).

MS-DOS, Windows, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows 2003, Windows Vista и Windows 7 являются зарегистрированными торговыми марками Корпорации Microsoft (Microsoft Corporation). Другими торговыми марками и зарегистрированными торговыми марками есть: SUSE и YaST компании SUSE Linux GmbH, Red Hat и Fedora являются зарегистрированными торговыми марками компании Red Hat, Inc. Mandrake является зарегистрированной торговой маркой компании MandrakeSoft, SA. Debian является зарегистрированной торговой маркой организации Software in the Public Interest, Inc. MySQL и логотип MySQL являются зарегистрированными торговыми марками корпорации Oracle Corporation и/или ее филиалов.

Все торговые марки используются без гарантий их свободного использования и, возможно, являются зарегистрированными торговыми марками.

OTRS AG, по существу, следует обозначениям производителей. Другие продукты, упомянутые в этом руководстве, могут быть торговыми марками соответствующих производителей.

Содержание

Список таблиц

Список примеров

Предисловие

Эта книга предназначена для использования OTRS-администраторами. Он также послужит хорошим справочником для новичков OTRS.

The following chapters describe the installation, configuration, and administration of the OTRS software. The first third of the text describes key functionality of the software, while the remainder serves as a reference to the full set of configurable parameters.

This book continues to be a work in progress, given a moving target on new releases. We need your feedback in order to make this a high quality reference document: one that is usable, accurate, and complete. Please write to us if you find content missing in this book, if things are not explained sufficiently, 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. Thanks in advance for your contributions!

Глава 1. Введение

1. Системы Обработки Заявок - Основы

This chapter offers a brief introduction to trouble ticket systems, along with an explanation of the core concept of a trouble ticket. A quick example illustrates the advantages of using such a system.

1.1. Что такое система обработки заявок и зачем она нужна?

Следующий пример показывает, что собой являет система обработки заявок и как можно сэкономить много времени и денег, используя подобную систему в своей компании.

Let's imagine that Max is a manufacturer of video recorders. Max receives many messages from customers needing help with the devices. Some days, he is unable to respond promptly or even acknowledge the messages. Some customers get impatient and write a second message with the same question. All messages containing support requests are stored in a single inbox folder. The requests are not sorted, and Max responds to the messages using a regular email program.

Since Max cannot reply fast enough to all the messages, he is assisted by the developers Joe and John in this. Joe and John use the same mail system, accessing the same inbox. They don't realize that Max often gets two identical requests from one frustrated customer. Sometimes they both end up responding separately to the same request, with the customer receiving two different answers. Furthermore, Max is unaware of the details of their responses. He is also unaware of the details of the customer problems and their resolutions, such as which problems occur with high frequency, or how much time and money he has to spend on customer support.

At a meeting, a colleague tells Max about trouble ticket systems and how they can solve Max's problems with customer support. After looking for information on the Internet, Max decides to install 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 automatically generates an 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. Customers are now happy because their requests are acknowledged and it is not necessary to send a second message with the same question. Max, John, and Joe can now log 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.

Давайте представим что г-н Гришко делает запрос в компанию Максима и его сообщение обрабатывается системой OTRS. Александр дает краткий ответ на его вопрос. Но у г-на Гришко есть дополнительный вопрос и на следующий день он отвечает на письмо Александра. Поскольку у Александра и своих дел хватает, на письмо г-на Гришко отвечает Максим. Способность OTRS хранить историю позволяет Максиму просмотреть всю цепочку переписки по этому запросу и написать более подробный ответ г-ну Гришко. Г-н Гришко, в свою очередь, даже не догадывается, что в процесс общения были вовлечены два разных человека, он доволен советами относительно решения своей проблемы, которые пришли в последнеем сообщении от Максима.

Of course, this is only a short preview of the possibilities and features of trouble ticket systems. But if your company has to attend to a high volume of customer requests through emails and phone calls, and if different service representatives need to respond at different times, a ticket system can be of great assistance. It can help streamline work flow processes, add efficiencies, and improve your overall productivity. A ticket system helps you to flexibly structure your Support or Help Desk environment. Communications between customers and service staff become more transparent. The net result is an increase in service effectiveness. And no doubt, satisfied customers will translate into better financial results for your company.

1.2. Что такое заявка?

Заявка очень похожа на медицинскую карточку пациента больницы. Медицинская карточка создается, когда пациент посещает больницу впервые, все важные данные о пациенте, такие как: личные данные, информация о состоянии его здоровья, медицинском осмотре записываются в медицинскую карточку. С каждым новым визитом пациента в больницу, каждый из лечащих врачей добавляет в медицинскую карточку информацию о состоянии пациента, истории болезни и, используемые для лечения лекарственные препараты. Таким образом другие лечащие врачи и медсестры могут видеть, подробную картину состояния пациента. Когда пациент выздоровел и выписался из больницы, медицинская карточка закрывается и вся информация передается в архив.

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 created. The response to this new ticket is comparable to a doctor's entry in the medical report. A ticket is closed if an answer is sent back to the customer, or if the ticket is separately closed by the system. If a customer responds again on an already closed ticket, the ticket is reopened with the new information added. Every ticket is stored and archived with complete information. Since tickets are handled like normal emails, attachments and contextual annotations will also be stored with each email. In addition, information on relevant dates, employees involved, working time needed for ticket resolution, etc. are also saved. At any later stage, tickets can be sorted, and it is possible to search through and analyze all information using different filtering mechanisms.

2. OTRS Help Desk

This chapter describes the features of OTRS Help Desk (OTRS). You will find information about the hardware and software requirements for OTRS. Additionally, in this chapter you will learn how to get commercial support for OTRS, should you require it, and how to contact the community.

2.1. Основы

OTRS Help Desk (OTRS) is a web application that is installed on a web server and can be used with a web browser.

OTRS is separated into several components. The main component is the OTRS framework which contains all central functions for the application and the ticket system. It is possible to install additional applications such as OTRS::ITSM modules, integrations with Network Monitoring solutions, a knowledge base (FAQ), et cetera.

2.2. Особенности и функциональные возможности

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

2.2.1. User interface

- OTRS comes with separate, modern web interfaces for agents and customers.

- It can be used on any modern web browser, including mobile plaforms and is retina ready.
- The web interface can be customized with own themes and skins.
- Powerful and customizable agent dashboard with personal ticket overviews and graphical statistics support.
- An extensible reporting engine provides various statistics and report scheduling options.
- With the ProcessManagement it is possible to define own ticket-based screens and processes (ticket workflows).
- OTRS has a built-in rights management that can be extended with fine-grained access control lists (ACLs).
- Support for more than 30 languages and different time zones.

2.2.2. Mail interface

- Support for MIME emails with attachments.
- Automatic conversion of HTML into plain text messages (increased security for sensitive content and enables faster searching).
- Incoming mail can be filtered and pre-processed with complex rules, e.g. for spam messages or Queue distribution.
- Support for PGP and S/MIME standards for key/certificate management and email processing.
- Automatic responses, configurable for every queue.
- Уведомления агентов по электронной почте о новых, последующих или разблокированых заявках.
- It is possible to define an own Ticket identifier to recognize follow-ups, e.g. Call#, Ticket# or Request#. There are different ticket number generators (date-based, random etc.) and you can integrate your own as well. Follow-ups can also be recognized by In-Reference-To headers or external ticket numbers.

2.2.3. Tickets

- OTRS uses Tickets to gather all external an internal communication that belongs together. These tickets are organized in Queues.
- There are many different ways of looking at the tickets in a system (based on Queues, Status, Escalation etc.) in different level of detail (small/medium/preview).
- The Ticket history records all changes to a ticket.
- Tickets can be changed in many ways, such as replying, forwarding, bouncing, moving to another Queue, updating attributes (state, priority etc.), locking and accounting working time. It is possible to modify many tickets at once (bulk action).
- Pending time and escalation time / SLA management allow time-based scheduling and restrictions on tickets.

- Tickets can be linked to other tickets or other objects such as FAQ entries.

- "GenericAgent" позволяет совершать автоматические и синхронные действия над заявками.

- OTRS comes with a powerful search engine that allows complex and fulltext searches on tickets.

2.2.4. System

- OTRS runs on many operating systems (Linux, Solaris, AIX, FreeBSD, OpenBSD, Mac OS 10.x, Microsoft Windows) and supports several database systems for the central OTRS back-end (MySQL, PostgreSQL, Oracle, MSSQL).

- The core system can be extended by installing OTRS packages. There are many free packages (such as FAQ, OTRS::ITSM and others) as well as FeatureAddon packages that are available for service contract customers of the OTRS group.

- Integration of external back-ends for the customer data, e.g. via AD, eDirectory or OpenLDAP. Customers can authenticate via database, LDAP, HTTPAuth or Radius.

- With the GenericInterface it is easy to connect OTRS to other web services. Simple web services can be integrated without programming, complex scenarios with custom extensions. The OTRS Ticket connector allows the creation, updating and searching of tickets, via web services from a third party application.

Now let us look at the changes in recent versions of OTRS.

2.2.5. New features of OTRS 3.3

2.2.5.1. Productivity

- Dashboard ticket lists and regular ticket overviews can now be filtered by eligible ticket columns, and the shown coumns are configurable.

- Ticket medium and preview overviews are now sortable.

- Added a calendar widget for the dashboard that can show tickets as events.

- Added new dashboard widget that shows in a matrix form the number of tickets per state and per queue.

- Agents can now mark important articles.

- A new tree selection widget makes working with tree data (queues, services etc.) much faster and easier.

- Added support to search relative dates (e.g. more than 1 month ago) in Date and Date/ Time dynamic fields.

- It is now possible to specify templates (previously "standard responses") also for creation of new tickets and for ticket forwarding.

- The list of available processes can now be filtered by ACLs.

- Added support to initiate processes from Customer Interface.

- In many places text is not shortened any more by a fixed number of characters ("Queue1..."), but instead by available screen estate. This makes it possible to see more information at once.
- OTRS is now Retina-ready. Images have been adapted to match the higher resolutions and most of the image icons have been replaced by font characters from the FontAwesome webfont.
- Added new feature "management dashboard". This makes it possible to display statistic charts in the dashboard. Please note that IE8 does not support this feature.

2.2.5.2. Working With External Systems

- OTRS can now use multiple customer company databases, thanks to Cyrille @ belnet-ict.
- OTRS can now automatically store customer user data in ticket dynamic fields for permanent storage in the ticket. This can be useful for reporting.
- OTRS is now able correctly assign incoming emails to existing tickets based on ticket numbers from external systems.
- OTRS can now fetch email also over POP3/TLS connections.

2.2.5.3. Installation & Administration

- Web Installer now can setup OTRS on PostgreSQL, Oracle and SQL Server databases in addition to MySQL.
- OTRS now has full support for MySQL 5.6.
- Generic agent jobs can now be executed for configured ticket events.
- The new graphical ACL editor makes ACL editing easier.
- Postmaster filters can now use negated filter conditions, thanks to Renée Bäcker.
- Postmaster filters can now specify relative pending dates and Owner / Responsible for new tickets based on incoming email data.
- Customer and Agent passwords now can be encrypted using the strong bcrypt algorithm, which is better than SHA.
- Many icons now use an icon font which makes it much easier to create custom skins with different base colors. This also improves overall performance through smaller amount of (image) files to load.

2.2.6. New features of OTRS 3.2

2.2.6.1. More customer focused

- The new "Customer Information Center" provides a great dashboard-like view on a customer (company). You can see
  - Escalated, reminder, new, and open tickets of the customer company.
  - Customer users (contacts) belonging to this customer company, with their individual ticket count and shortcuts for creating new tickets for them.

- An overall ticket status view of the customer company.

- New "switch to customer" feature makes it possible for an agent with the required permissions to look into the customers's panel with their rights.

2.2.6.2. More customizable

2.2.6.2.1. Управление Процессами

- The new process management makes it possible to represent processes within OTRS.

2.2.6.2.2. Customer Interface Improvements

- The customer web interface now fully supports AJAX and ACLs.

- It now requires JavaScript and is not compatible with Internet Explorer 6 or earlier versions.

- In the Customer Interface, you can now set the default ticket type for new tickets. Additionally, you can now also hide the ticket type and use a default value for all tickets created via the customer interface.

2.2.6.2.3. Agent Interface Improvements

- Agents can now search for tickets based on escalation time.

- New option to show DynamicFields by default in ticket search.

- Screen usage optimizations in the ticket screens to avoid scrolling in popup windows. For each ticket screen, the size of the richtext editor can now be configured separately.

- It is now possible to move tickets to another queue from within the TicketAction dialogs (TicketNote, TicketClose etc.) after activating a configuration option. This is turned off by default.

- Ticket search will now directly jump to the ticket zoom screen if only one ticket is found.

- New ability to hide the Article Type from TicketActionCommon-based screens which can be helpful to fit more data in the browser window.

- There is a new out-of-office dashboard widget that lists all currently unavailable agent colleagues.

- New CKEditor 4 makes working with rich text content (such as HTML emails) easier and more stable.

2.2.6.2.4. Administration improvements

- Event Based notifications can now be sent out only for specific Article Sender Types.

- The Statistics engine in OTRS now understands 'Weeks' in addition to days, months and years. This grants the ability to, for instance, create a report for tickets 'created last week', or generate a report that shows tickets created per queue per week.

- It is possible to place customized DTL (template) files in `Custom/Kernel/Output/HTML`, so that they override the system's default DTL files just as how this already works for Perl files.

- In AdminSMIME it is now possible to display human readable certificate contents.
- SysConfig now supports config setting types Date and DateTime.

2.2.6.3. Better scalability

2.2.6.3.1. Ticket Archiving Improved

- When tickets are archived, the information which agent read the ticket and articles can be removed, as well as the ticket subscriptions of agents. This is active by default and helps reduce the amount of data in the database on large systems with many tickets and agents.
- There is also a new script to remove this data from existing archived tickets.
- Archived tickets are now always shown as 'read' by the agent.

2.2.6.3.2. Performance Improvements

- Session management is up to 10 times faster, especially with many active users.
- It is now possible to limit the number of concurrent agents and/or users to avoid server capacity overload.
- Significant reduction in the number of executed database statements in ticket overviews and ticket masks in agent and customer frontend.
  - This will reduce the load on database servers, especially on large systems. In some cases OTRS will become visibly more responsive (if the system was slowed down by the DB load or latency).
- Improved performance of LDAP user synchronization.
- Improved cache performance with many cache files.

2.2.6.4. More Interoperable

2.2.6.4.1. FAQ Connector for the GenericInterface

- It is now possible to access the data of the FAQ module (OTRS knowledge database) via web service (GenericInterface). This can be useful to embed FAQ articles on your company website, for example.

2.2.7. Новые возможности OTRS 3.1

2.2.7.1. GENERIC INTERFACE - A Web Service Framework

- GI гибкий фреймворк, который позволяет веб-сервисам OTRS взаимодействовать с приложениями сторонних производителей.
- OTRS может работать как поставщик (сервер, который получает запросы от удаленной стороны) или же как клиент (отправлять запросы).
- Просты подключения к веб-сервисам могут быть созданы путем конфигурации Generic Interface, не требуя дополнительного программирования.
- Сложные сценарии могут быть реализованы путем подключения пользовательских расширений OTRS, которые добавляют perl-код на различных архитектурных уровнях.

- *Connectors* expose parts of OTRS to Generic Interface web services.For example, a ticket connector exposes the ticket create/update function, so that they can be used in a web service regardless which network transport is used.
- A scheduler daemon process supports asynchronous event handling. This is useful to asynchronously start web service requests from OTRS to another system, after the agent''s request has been answered (e.g. when a ticket has been created). Otherwise, it might block the response, resulting in increased response times for the agent.

With the Generic Interface, new web services can be configured easily by using existing OTRS modules, without additional code. They can be combined to create a new web service. When configuring a new web service connection, the administrator has to add:

- Новый веб-сервис в GUI администратора
- Основные мета-данные (транспортные (SOAP), URL и т.д.) и
- Существующие операции (часть коннектора) и задать для каждой операции правила, как должны быть сопоставлены данные (входящие и исходящие)

Отладчик Generic Interface поможет OTRS-администратору проверить как поступают запросы и как они обрабатываются в рамках различых слоев.

2.2.7.1.1. Current Features

- Сетевые транспортные протоколы: SOAP/HTTP. Поддержка протоколов REST and JSON будет добавлена в будущем, в зависимости от спроса потребителей.
- Настраиваемое отображение данных в GUI для преобразования key/value-значений входящих и исходящих данных.
- Графический отладчик для проверки конфигурации и потока информации от настроеных веб-сервисов.
- Ticket-коннектор позволяет использовать OTRS для обработки заявок как веб-сервис.

2.2.7.1.2. Future Features

- Дополнительные сетевые транспортные протоколы (REST, JSON).
- GI заменит iPhoneHandle как backend для мобильных приложений.
- Additional connectors will be added to provide more parts of OTRS for use with web services (e.g. to allow the creation, update, or deletion of agents, users, services or CIs).

2.2.7.2. DYNAMIC FIELDS

Динамические поля (DynamicFields) заменяют существующие поля FreeText и FreeTime с динамической структурой для заявок и статей. Они также позволяют создавать пользовательские формы в OTRS.

- Неограниченное количество полей можно настроить с помощью собственного графического пользовательского интерфейса администрирования.
- Поля могут иметь различные типы, которые могут быть использованы как для заявок так и для статей. По умолчанию доступны:
  - Text

- Multiline text

- Checkbox

- Dropdown

- Multi-select

- Date

- Дата и время

- Новые пользовательские типы полей (например выпадающий список, который получает данные из внешнего источника) могут быть добавлены с небольшими усилиями, поскольку поля создаются по модульному принципу.

- A future scenario is, that DynamicFields can be used for objects other than tickets or in custom modules. For example, a custom module adding objects to handle "orders" in OTRS could use the DynamicFields to attach properties/data to these orders.

- Скрипт обновления базы данных превратит поля FreeText и связанные с ними настройки в новую структуру.

2.2.7.3. TICKET MANAGEMENT IMPROVEMENTS

2.2.7.3.1. Ticket creation improved

- Когда создается новая заявка на основе email или телефонного звонка, множество email-адресов может быть задано как 'To:', 'CC:' or 'BCC:' .

2.2.7.3.2. Inbound phone call support

- Теперь для созданых заявок могут быть зарегистрированы входящие телефонные звонки (до этого, можно было регистрировать только исходящие звонки).

2.2.7.3.3. Ticket overview preview improved

- Теперь в SysConfig можно исключать статьи отдельных типов (напримр, статьи от внутренних агентов), это будет предотвращать их появление в режиме предварительного просмотра.

- Определенный тип статьи может быть настроен таким образом, что будет отображать статьи этого типа полностью, если имеются соответствующие права для их просмотра.

2.2.7.3.4. Ticket move improved

- Страница, которая отображается после перемещения заявки теперь может быть настроена пользователем. Варианты просмотра: просмотр расширенной информации (LastScreenView) или же список заявок (LastScreenOverview).

2.2.7.3.5. Bulk action improved

- With the new bulk action, outbound emails can now be sent from multiple tickets at the same time. As tickets can have different queues, and these queues each can have different templates, salutations and signatures, these are not used in the Bulk Action email.

- Дополнительные массовые действия позволяют настроить тип заявки для отдельных заявок.

2.2.7.3.6. Configurable Reject Sender Email Address

- The feature allows configuring an email address instead of the administrator address to reject the creation of new tickets by email. This feature can be used in all cases where customers are not allowed to create new tickets by email.

2.2.7.4. PROCESS AUTOMATION

2.2.7.4.1. Escalation events added

- OTRS will now create events for each of the available escalation types (response, update and resolution). This allows performing actions (such as notifications) before the escalation occurs, in the moment it occurs, and in the moment that the escalation ends.

2.2.7.4.2. Notification mechanism improved

- Новый generic-agent-модуль уведомлений позволяет OTRS-администратору определить сообщения, которые будут показаны в веб-интерфейсе агентов, когда они войдут в систему.

2.2.7.4.3. Time calculation improved

- All kinds of times will now be calculated by and based on the application server only solving the issues that were caused by variances between the clock times of application and data base servers.

2.2.7.4.4. GenericAgent improved

- Теперь GenericAgent можеть фильтровать заявки по времени изменения.

- Кроме того, GenericAgent может установить ответственность за заявку для схожих (соответствующих) заявок.

2.2.7.5. USER INTERFACE, RICH TEXT EDITOR, CHARSET

2.2.7.5.1. User interface performance improved

- Улучшена скорость рендеринга и отображения статей, большое спасибо Stelios Gikas <stelios.gikas@noris.net>!

2.2.7.5.2. Rich Text Editor Update

- Добавлена поддержка IOS5

- Блок цытат можно оставить с помощью клавиши ввода.

- Обновлен CKEditor 3.4 до CKEditor 3.6, про улучшения можно прочесть перейдя по ссылкеCKEditor 3.5 and CKEditor 3.6.

- Улучшена поддержка браузера IE9.

- Диалоговые окна изменяемого размера.

2.2.7.5.3. Unicode Support - Non-UTF-8 Internal Encodings Dropped

- UTF-8 теперь единственная внутренняя кодировка, разрешенная OTRS.

- Теперь все языковые файлы отформатированы в кодировке UTF-8, что значительно упрощает их обработку и дальнейшее усовершенствование механизма перевода.

2.2.7.6. DATABASE DRIVER SUPPORT

2.2.7.6.1. PostgreSQL DRIVER compatibility improved

- Добавлена поддержка PostgreSQL 9.1

- Доступен новый драйвер для PostgreSQL 8.1 или более ранних версий.

2.2.7.6.2. MS SQL DRIVER compatibility improved

- Теперь драйвер MS SQL сохраняет бинарные данные используя тип данных VARBINARY а не устаревший тип TEXT, тип NVARCHAR также используется вместо устаревшего VARCHAR (это делается для улучшенной поддержки Unicode).

2.2.7.7. MAIL INTEGRATION

2.2.7.7.1. Mail handling improved

- When connecting to IMAP mail accounts, it is now possible to handle emails from a specific email folder, aside from the INBOX folder.

- OTRS can now connect to IMAP servers using Transport Layer Security (TLS), which is useful for modern restricted environments.

2.3. Требования к апаратному и програмному обеспечению

OTRS can be installed on many different operating systems. OTRS can run on linux and on other unix derivates (e.g. OpenBSD or FreeBSD). You can also run it on Microsoft Windows. OTRS does not have 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 for a small setup.

To run OTRS, you'll also need to use a web server and a database server. Apart from that, you should install perl and/or install some additional perl modules on the OTRS machine. The web server and Perl must be installed on the same machine as OTRS. The database back-end may be installed locally or on another host.

For the web server, we recommend using the Apache HTTP Server, because its module mod_perl greatly improves the performance of OTRS. Apart from that, OTRS should run on any web server that can execute Perl scripts.

You can deploy OTRS on different databases. You can choose between MySQL, PostgreSQL, Oracle, or Microsoft SQL Server. If you use MySQL you have the advantage that the database and some system settings can be configured during the installation, through a web front-end.

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

Требования к програмному обеспечению

2.3.1. поддержка Perl

- Perl 5.10 or higher

2.3.2. Поддержка Веб-сервера

- Apache2 + mod_perl2 or higher (recommended)
- Веб-сервер с поддержкой CGI-интерфейса (CGI-интерфейс не рекомендуется)
- Microsoft Internet Information Server (IIS) 6 или выше

2.3.3. Поддержка базы данных

- MySQL 5.0 or higher
- PostgreSQL 8.0 or higher (8.2 or higher recommended)
- Oracle 10g или выше
- Microsoft SQL Server 2005 или выше

Раздел руководства установка Perl-модулей более подробно описывает как можно настроить модули, необходимые для работы OTRS.

Если для установки OTRS используются бинарные пакеты, собраные специально под вашу операционную систему, то все необходимые модули языка Perl уже включены. Если нет, - используйте менеджер пакетов вашей операционной системы (например (rpm, Windows-Инсталлятор) для установки всех необходимых модулей.

2.3.4. Поддержка веб-браузера

To use OTRS, you'll be OK if you use a modern browser with JavaScript support enabled. We do not support the following browsers:

- Internet Explorer before version 8
- Firefox before version 10
- Safari before version 5

We recommend always using the latest version of your browser, because it has the best JavaScript and rendering performance. Dramatical performance varieties between the used browsers can occur with big data or big systems. We are happy to consult you on that matter.

2.4. Комюнити

OTRS has a large user community. Users and developers discuss OTRS and exchange information on related issues through the mailing-lists. You can use the mailing lists to discuss installation, configuration, usage, localization and development of OTRS. You can report software bugs in our bug tracking system.

Домашняя страница OTRS-сообщества: http://www.otrs.com/open-source/.

2.5. Professional Services for OTRS

Whether you need help in configuring or customizing OTRS or you want to be on the safe side, don't hesitate to contact us: We offer a wide range of professional services such as world-wide enterprise support, consulting and engineering including process design, implementation, customization, application support, and fully managed service.

Our Service Contracts guarantee instant help and professional support as well as support assessment and last but not least free access to OTRS Feature Add-ons - useful additional features for your OTRS.

The OTRS Group offers specific training programs in different countries. You can either participate in one of our public OTRS Administrator trainings which take place regularly, or benefit from an inhouse training that covers all the specific needs of your company.

Глава 2. Установка

В этой главе описывается установка и базовая конфигурация главного OTRS-фреймворка. Здесь вы найдете информацию об установке OTRS из исходных кодов или из бинарных пакетов, например RPM, или с помощью Windows-инсталлятора.

В этой главе рассматриваются такие вопросы как: конфигурация веб-сервера и сервера базы данных, интефейса между OTRS и базой данных, установка дополнительных Perl-модулей, установка соответствующих прав доступа для OTRS, настройка планировщика задач cron jobs для OTRS и основных параметров в конфигурационных файлах.

Следуйте подробным шагам этой главы чтобы установить OTRS на своем сервере. Потом можно использовать его веб-интерфейс чтобы войти в систему и производить администрирование.

1. Самый простой способ - установка из готовых бинарных пакетов

If available for your platform you should use pre-built packages to install OTRS, since it is the simplest and most convenient method. You can find them in the download area at http://www.otrs.com . The following sections describe the installation of OTRS with a pre-built or binary package on SUSE, Red Hat and Microsoft Windows systems. Only if you are unable to use the pre-built packages for some reason should you follow the manual process.

1.1. Установка из rpm-пакетов на сервер под управлением Suse Linux

This section describes the installation of our RPM package on a SUSE Linux server. We have tested against all recent SLES and openSUSE versions. Before you start the installation, please visit http://www.otrs.com/downloads and make sure you use the latest OTRS RPM package available.

1.1.1. Preparing the database for OTRS

You can use OTRS using different database back-ends: MySQL, PostgreSQL, Oracle or Microsoft SQL Server. The most popular database to deploy OTRS on is MySQL. This chapter shows the steps you need to take to configure MySQL on a SUSE-based server. Of course you can install the database on a dedicated database server if needed for scalability or other purposes.

Примечание

If you follow this chapter on openSUSE 12.3 and up you'll actually not install MySQL but MariaDB instead, a MySQL compatible fork of the MySQL code. This is no problem, it will work just as well (and even a little better at some points).

Install MySQL by executing the following command as root:

```
linux:~ # zypper install mysql perl-DBD-mysql
```

This will install MySQL with the default options on your system. You'll need to change the defaults in order to make it suitable for OTRS. With a text editor open the file `/etc/my.cnf` and change the line with **max_allowed_packet** on it, and add a line below, like this:

```
max_allowed_packet=20M
query_cache_size=32M
```

Now execute **rcmysql restart** to re-start the database server and activate these changes. Then run **/usr/bin/mysql_secure_installation** and follow the on-screen instructions to set a database root password, remove anonymous access and remove the test database. Lastly, run **chkconfig -a mysql** in order to make sure mysql is automatically started at server startup time.

1.1.2. Installing OTRS

Install OTRS with via the command line using **zypper**. This will also pull in some dependencies such as the apache web server and some Perl modules. Make sure you copied the OTRS RPM file to the current directory.

```
otrs-sles:~ # zypper install otrs-3.3.*.rpm
....
Retrieving package otrs-3.3.3-01.noarch (1/26), 17.5 MiB (74.3 MiB unpacked)
Installing: otrs-3.3.3-01 [done]
Additional rpm output:
Check OTRS user ... otrs added.

Next steps:

[start database and Apache]
 Make sure your database is running and execute 'rcapache2 restart'.

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

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

       stop-force}).

((enjoy))

 Your OTRS Team
 http://otrs.org/

otrs-sles:~ #
```

Сценарий: Команда для установки OTRS.

The OTRS installation is done. Start your web server to load the OTRS specific changes in its configuration, as shown in the script below. Also run **chkconfig** to make sure OTRS is automatically started when the server reboots.

```
otrs-sles:~ # chkconfig -a apache2
apache2                   0:off  1:off  2:off  3:on   4:off  5:on   6:off
otrs-sles:~ # rcapache2 start
Starting httpd2 (prefork) httpd2-prefork: Could not reliably determine the server's fully
 qualified domain name, using 10.x.x.x for ServerName

                                                                                           done
otrs-sles:~ #
```

Script: Starting the web server.

1.1.3. Installation of additional perl modules

OTRS needs some more modules than can be installed by the RPM. You can post-install them manually. You can check what modules you are missing by running the bin/

otrs.CheckModules.pl script located in the /opt/otrs directory. Some modules are only needed for optional functionality, such as communication with IMAP(S) servers or PDF generation. On SLES you should add an external repository to the zypper configuration in order to get the modules needed for your system. Choose the module needed for your OS version from here: http://download.opensuse.org/repositories/devel:/languages:/perl/. Add the repository like this for SLES 11 SP2:

```
            zypper ar -f -n perl http://download.opensuse.org/repositories/devel:/languages:/
perl/SLE_11_SP2 Perl
```

On openSUSE 12.3 the extra repository is only needed for the Mail::IMAPClient module, which you'd only need if you need to collect mails from an IMAP server secured with TLS. The corresponding line would look like this:

```
            zypper ar -f -n perl http://download.opensuse.org/repositories/devel:/languages:/
perl/openSUSE_12.3/ Perl
```

The first time you use zypper after you added this repository, you will be prompted to add its key. Now you can install missing modules like below.

```
otrs-sles:/opt/otrs # zypper install -y "perl(YAML::LibYAML)"
Refreshing service 'susecloud'.
Retrieving repository 'perl' metadata [\]

New repository or package signing key received:
Key ID: DCCA98DDDCEF338C
Key Name: devel:languages:perl OBS Project <devel:languages:perl@build.opensuse.org>
Key Fingerprint: 36F0AC0BCA9D8AF2871703C5DCCA98DDDCEF338C
Key Created: Wed Oct 10 22:04:18 2012
Key Expires: Fri Dec 19 22:04:18 2014
Repository: perl

Do you want to reject the key, trust temporarily, or trust always? [r/t/a/?] (r): a
Retrieving repository 'perl' metadata [done]
Building repository 'perl' cache [done]
Loading repository data...
Reading installed packages...
'perl(YAML::LibYAML)' not found in package names. Trying capabilities.
Resolving package dependencies...

The following NEW package is going to be installed:
  perl-YAML-LibYAML

The following package is not supported by its vendor:
  perl-YAML-LibYAML

Retrieving package perl-YAML-LibYAML-0.38-12.4.x86_64 (1/1), 75.0 KiB (196.0 KiB unpacked)
Retrieving: perl-YAML-LibYAML-0.38-12.4.x86_64.rpm [done (55.7 KiB/s)]
Installing: perl-YAML-LibYAML-0.38-12.4 [done]
```

The next step is to configure OTRS using the web installer, as described in this section.

1.2. Installing OTRS on a Red Hat Enterprise Linux or CentOS system

This section describes the installation of our RPM package on a Red Hat Enterprise Linux (RHEL) or CentOS server. For OTRS 3.3 and up, RHEL 6 or CentOS 6 is a requirement, version 5 is not supported. Before you start the installation, please visit http://www.otrs.com/downloads and make sure you use the latest OTRS RPM package available.

1.2.1. Preparing the database for OTRS

You can use OTRS using different database back-ends: MySQL, PostgreSQL, Oracle or Microsoft SQL Server. The most popular database to deploy OTRS on is MySQL. This chapter shows the steps you need to take to configure MySQL on a RHEL-based server. Of course you can install the database on a dedicated database server if needed for scalability or other purposes.

Install MySQL by executing the following command as root:

```
        [root@otrs-centos6 ~]# yum -y install mysql-server
```

This will install MySQL with the default options on your system. You'll need to change the defaults in order to make it suitable for OTRS. With a text editor open the file `/etc/my.cnf` and add the next two lines under the [mysqld] section:

```
        max_allowed_packet=20M
        query_cache_size=32M
```

Now execute **service mysqld start** to re-start the database server and activate these changes. Then run **/usr/bin/mysql_secure_installation** and follow the on-screen instructions to set a database root password, remove anonymous access and remove the test database. Lastly, run **chkconfig mysqld on** in order to make sure mysql is automatically started at server startup time.

1.2.2. Installing OTRS

Install OTRS with via the command line using **yum**. This will also pull in some dependencies such as the apache web server and some Perl modules. Make sure you copied the OTRS RPM file to the current directory.

```
[root@otrs-centos6 ~]# yum install --nogpgcheck otrs-3.3.*.rpm
...
Dependencies Resolved

================================================================================
 Package                     Arch   Version                Repository       Size
================================================================================
Installing:
 otrs                        noarch 3.3.3-01               /otrs-3.3.3-01.noarch
                                                                           74 M
Installing for dependencies:
 apr                         x86_64 1.3.9-5.el6_2          updates         123 k
 ...
 procmail                    x86_64 3.22-25.1.el6          base            163 k

Transaction Summary
================================================================================
Install      26 Package(s)

Total size: 80 M
Total download size: 6.0 M
Installed size: 88 M
Downloading Packages:
(1/25): apr-1.3.9-5.el6_2.x86_64.rpm                     | 123 kB     00:00
...
(25/25): procmail-3.22-25.1.el6.x86_64.rpm               | 163 kB     00:00
--------------------------------------------------------------------------------
Total                                           887 kB/s | 6.0 MB     00:06
```

```
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : apr-1.3.9-5.el6_2.x86_64                               1/26
  ...
  Installing : otrs-3.3.3-01.noarch                                   26/26
Check OTRS user ... otrs added.

Next steps:

[httpd services]
 Restart httpd 'service httpd restart'

[install the OTRS database]
 Make sure your database server is running.
 Use a web browser and open this link:
 http://myserver.example.com/otrs/installer.pl

[OTRS services]
 Start OTRS 'service otrs start' (service otrs {start|stop|status|restart).

((enjoy))

 Your OTRS Team

Installed:
  otrs.noarch 0:3.3.3-01

Dependency Installed:
  ...

Complete!
[root@otrs-centos6 ~]#
```

Сценарий: Команда для установки OTRS.

The OTRS installation is complete. Now you should make sure that Apache is started and that it starts whenever the server reboots.

```
[root@otrs-centos6 ~]# chkconfig httpd on
[root@otrs-centos6 ~]# service httpd start
Starting httpd: httpd: Could not reliably determine the server's fully qualified domain name,
 using 10.x.x.x for ServerName [  OK  ]
[root@otrs-centos6 ~]#
```

Script: Starting the web server.

1.2.3. Installation of additional perl modules

OTRS needs some more modules than can be installed by the RPM. You can post-install them manually. You can check what modules you are missing by running the `bin/otrs.CheckModules.pl` script located in the `/opt/otrs` directory. Some modules are only needed for optional functionality, such as communication with IMAP(S) servers or PDF generation. On Red Hat or CentOS we recommend installing these modules from the EPEL repository, a repository maintained by the Fedora project, which provides high quality packages for RHEL and derivatives. Check for more information the EPEL web site.

If you're on RHEL 6 or CentOS 6, you can get the latest package for EPEL from this site. You can add this repository to yum it in one go by copying the RPM URL you find on this page and executing this command:

```
 [root@otrs-centos6 otrs]# yum -y install http://download.fedoraproject.org/pub/epel/6/i386/
epel-release-6-8.noarch.rpm
Loaded plugins: security
Setting up Install Process
epel-release-6-8.noarch.rpm                                   |  14 kB     00:00
Examining /var/tmp/yum-root-7jrJef/epel-release-6-8.noarch.rpm: epel-release-6-8.noarch
Marking /var/tmp/yum-root-7jrJef/epel-release-6-8.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package epel-release.noarch 0:6-8 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====================================================================================
 Package             Arch          Version     Repository                        Size
=====================================================================================
Installing:
 epel-release        noarch        6-8         /epel-release-6-8.noarch         22 k

Transaction Summary
=====================================================================================
Install       1 Package(s)

Total size: 22 k
Installed size: 22 k
Downloading Packages:
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : epel-release-6-8.noarch                                           1/1
  Verifying  : epel-release-6-8.noarch                                           1/1

Installed:
  epel-release.noarch 0:6-8

Complete!
[root@otrs-centos6 otrs]#
```

The first time you use yum after you added this repository, you will be prompted to add its key. Now you can install missing modules like below.

```

[root@otrs-centos6 otrs]# yum -y install "perl(Text::CSV_XS)"
Loaded plugins: security
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package perl-Text-CSV_XS.x86_64 0:0.85-1.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====================================================================================
 Package                   Arch            Version               Repository     Size
=====================================================================================
Installing:
 perl-Text-CSV_XS          x86_64          0.85-1.el6            epel           71 k

Transaction Summary
=====================================================================================
Install       1 Package(s)

Total download size: 71 k
Installed size: 154 k
```

```
Downloading Packages:
perl-Text-CSV_XS-0.85-1.el6.x86_64.rpm                        |  71 kB     00:00
warning: rpmts_HdrFromFdno: Header V3 RSA/SHA256 Signature, key ID 0608b895: NOKEY
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6
Importing GPG key 0x0608B895:
 Userid : EPEL (6) <epel@fedoraproject.org>
 Package: epel-release-6-8.noarch (@/epel-release-6-8.noarch)
 From   : /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6
Is this ok [y/N]: y
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : perl-Text-CSV_XS-0.85-1.el6.x86_64                                1/1
  Verifying  : perl-Text-CSV_XS-0.85-1.el6.x86_64                                1/1

Installed:
  perl-Text-CSV_XS.x86_64 0:0.85-1.el6

Complete!
[root@otrs-centos6 otrs]#
```

The next step is to configure OTRS using the web installer, as described in this section.

1.2.4. Installation of Oracle database driver on Red Hat / CentOS

If you want to deploy OTRS on an Oracle database, you'll need to compile and install the DBD::Oracle database driver. This is slightly more complicated than installing any of the other packages; this is because Oracle is a proprietary database and Red Hat nor the CentOS project are allowed to distribute drivers in their RPM repositories.

First of all, we'd need to install gcc, make and CPAN so we can compile and install the driver. Below you see the command on CentOS; on other versions it might look a little different.

```
[root@otrs-centos6 otrs]# yum -y install gcc make "perl(CPAN)"
```

The next step is to obtain and install the database client. For this you would need to sign up for a free account at the Oracle website. You can download the drivers from this page: http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html Please choose the Linux x86 or x86-64 version corresponding to the architecture of your system. You can check this with the **uname -i**. It is either 'x86_64' for x86-64 or 'i386' for x86. You should download the packages 'Instant Client Package - Basic', 'Instant Client Package - SQL*Plus', and 'Instant Client Package - SDK'. Save them to a location on your disk. Now as the root user you can install the packages using the following command:

```
[root@otrs-centos6 otrs]# yum install oracle-instantclient*
```

After this you should set two environment variables and compile the DBD::Oracle driver. Again, perform these tasks as the root user. The steps are outlined below. Please note that for briefness some lines outputted by the commands have been removed.

```
[root@otrs-centos6 otrs]# export ORACLE_HOME=/usr/lib/oracle/11.2/client64
[root@otrs-centos6 otrs]# export LD_LIBRARY_PATH=$ORACLE_HOME/lib
[root@otrs-centos6 otrs]# cpan
cpan[1]> look DBD::Oracle
...
Fetching with LWP:
  http://www.perl.org/CPAN/authors/id/P/PY/PYTHIAN/CHECKSUMS
```

```
Checksum for /root/.cpan/sources/authors/id/P/PY/PYTHIAN/DBD-Oracle-1.62.tar.gz ok
Scanning cache /root/.cpan/build for sizes
DONE
...
Working directory is /root/.cpan/build/DBD-Oracle-1.62-ZH6LNy
[root@localhost DBD-Oracle-1.62-ZH6LNy]# perl Makefile.PL
...
[root@localhost DBD-Oracle-1.62-ZH6LNy]# make
...
[root@localhost DBD-Oracle-1.62-ZH6LNy]# make install
...
cpan[2]> exit
Terminal does not support GetHistory.
Lockfile removed.
```

Now you should edit the file Kernel/Config.pm to provide ORACLE_HOME. The next step is to configure OTRS using the web installer, as described in this section.

1.3. Installing OTRS on a Debian or Ubuntu system

Важно

Please install OTRS from source, and do not use the OTRS packages that Debian/Ubuntu provides.

The installation of required Perl modules is easier if you use the available packages:

```
apt-get install libapache2-mod-perl2 libdbd-mysql-perl libtimedate-perl libnet-dns-perl \
    libnet-ldap-perl libio-socket-ssl-perl libpdf-api2-perl libdbd-mysql-perl libsoap-lite-perl
 \
    libgd-text-perl libtext-csv-xs-perl libjson-xs-perl libgd-graph-perl libapache-dbi-perl
```

1.4. Установка OTRS на операционной системе Microsoft Windows

1.4.1. Using the Windows Installer

Installing OTRS on a Microsoft Windows system is very easy. Download the latest installer for Win32 from http://www.otrs.com/try/ and save the file to your local file system. Then simply double-click on the file to execute the installer, and follow the few installation steps to setup the system. The installation wizard will take care of installing all components for OTRS. The next step is to configure OTRS using the web installer, as described in this section.

Важно

The Windows installer for OTRS contains all needed components for OTRS, i.e. the Apache web server, the MySQL database server, Perl (with all needed modules) and cron for Windows. However you can also use it to set up OTRS on the Microsoft IIS Web Server instead of on Apache. Also you can choose to not let it install the bundled MySQL, for instance if you want to deploy on a central MySQL database; a PostgreSQL database, on Microsoft SQL Server or on Oracle.

1.4.2. Installation of Oracle database driver on Windows

If you install OTRS using the Windows Installer, it will automatically configure all Perl modules you will typically need. There is one exception to this: if you need to deploy OTRS on an Oracle database, you'll need to post-install the Oracle driver. This is needed because there is no open source driver for Oracle available; we are not allowed to redistribute it in the OTRS package. There are two different perls you can use on Windows; you can choose the bundled Strawberry Perl, which is the default option and is recommended for proof-of-concept setups;

or you can choose Active State perl which is a separate download and install. This last option is recommended for production setups.

1.4.2.1. Installation of DBD::Oracle on ActiveState Perl

If you have installed ActiveState Perl it already comes with the Perl database driver DBD::Oracle only this should be linked to the Oracle C client. OTRS only works on 32-bit ActiveState Perl. This means you should also download the 32-bit Oracle client. Navigate to the Oracle client web site *Instant Client Downloads for Microsoft Windows (32-bit)* and choose 'Instant Client Package - Basic'. You'll need to sign up for a free Oracle Web account in order to download the software. Extract the download, a zip file, to the root of your drive, for instance `C:\instantclient_11_2`.

Now you should add the client libraries to your path and set an environment variable; 'ORACLE_HOME'. Navigate to Control Panel > System and Security > System, choose 'Advanced System Settings', click on 'New' under 'System variables', enter 'ORACLE_HOME' as Variable Name and use 'C:\instantclient_11_2' as the Variable Value. Select the 'PATH' variable, and append the directory 'C:\instantclient_11_2' to the path. If you now start a new cmd.exe session and run the otrs.CheckModules.pl script located in the OTRS directory, you should see that the DBD::Oracle driver is successfully installed. After this, you should restart the system in order for the other applications to detect the new environment variables.

1.4.2.2. Installation of DBD::Oracle on Strawberry Perl

In order to compile the DBD::Oracle module, you need to have installed the components of the OTRS web installer in a path without spaces. It should NOT be in `C:\Program Files\OTRS` but it should rather be in `C:\OTRSComponents` or similar.

On Strawberry Perl you should compile the Oracle driver manually. The OTRS Windows Installer bundles 32-bit Strawberry Perl. This means you should also download the 32-bit Oracle client. Navigate to the Oracle client web site *Instant Client Downloads for Microsoft Windows (32-bit)* and download the packages 'Instant Client Package - Basic', 'Instant Client Package - SQL*Plus', and 'Instant Client Package - SDK'. You'll need to sign up for a free Oracle Web account in order to download the software. Extract the downloads, all zip files, to the root of your drive, for instance `C:\instantclient_11_2`.

Now you should add the client libraries to your path and set an environment variable; 'ORACLE_HOME'. Navigate to Control Panel > System and Security > System, choose 'Advanced System Settings', click on 'New' under 'System variables', enter 'ORACLE_HOME' as Variable Name and use 'C:\instantclient_11_2' as the Variable Value. Select the 'PATH' variable, and append the directory 'C:\instantclient_11_2' to the path. After this, you should restart the system in order for the other applications to detect the new environment variables.

Now you can compile and install the DBD::Oracle module. For this, open a cmd.exe window, and start the cpan client. Download and extract DBD::Oracle via **look DBD::Oracle**, then type **perl Makefile.PL**, **dmake** and **dmake install**. Sample output is below.

```
C:\Users\max>cpan

cpan shell -- CPAN exploration and modules installation (v1.9800)
Enter 'h' for help.


cpan> look DBD::Oracle
Database was generated on Thu, 16 May 2013 14:52:41 GMT
Running look for module 'DBD::Oracle'

Trying to open a subshell in the build directory...
```

```
....
C:\OTRSCO~1\StrawberryPerl\cpan\build\DBD-Oracle-1.62-kq6los>perl Makefile.PL
Using DBI 1.623 (for perl 5.016003 on MSWin32-x86-multi-thread) installed in C:/
OTRSComponents/StrawberryPerl/perl/vendor/lib/auto/DBI/

Configuring DBD::Oracle for perl 5.016003 on MSWin32 (MSWin32-x86-multi-thread)

...
Checking if your kit is complete...
Looks good
LD_RUN_PATH=c:/instantclient_11_2/lib:c:/instantclient_11_2/rdbms/lib
Using DBD::Oracle 1.62.
Using DBD::Oracle 1.62.
Using DBI 1.623 (for perl 5.016003 on MSWin32-x86-multi-thread) installed in C:/
OTRSComponents/StrawberryPerl/perl/vendor/lib/auto/DBI/
Writing Makefile for DBD::Oracle
Writing MYMETA.yml and MYMETA.json
...

C:\OTRSCO~1\StrawberryPerl\cpan\build\DBD-Oracle-1.62-kq6los>dmake
... lots of output and compiler warnings ...

C:\OTRSCO~1\StrawberryPerl\cpan\build\DBD-Oracle-1.62-kq6los>dmake install
Files found in blib\arch: installing files in blib\lib into architecture depende
nt library tree
Installing C:\OTRSCO~1\StrawberryPerl\perl\site\lib\auto\DBD\Oracle\dbdimp.h
...
Appending installation info to C:\OTRSCO~1\StrawberryPerl\perl\lib/perllocal.pod

C:\OTRSCO~1\StrawberryPerl\cpan\build\DBD-Oracle-1.62-kq6los>exit

cpan> quit
Lockfile removed.

C:\Users\max>
```

1.4.2.3. Configure OTRS using the Web Installer

The web installer may be used to configure basic OTRS settings as described in this section.

2. Установка из исходных кодов (Linux, Unix)

2.1. Подготовка к установке из исходных кодов

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.com/try/

Распакуйте архив (например с помощью команды **tar**) в каталог `/opt` а затем переименуйте каталог otrs-x.x.x на otrs (см. ниже Сценарий ниже).

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

Script: Установка OTRS, первые шаги.

OTRS should NOT be run with root rights. Next you should add a new user for OTRS. 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 will need to add the new 'otrs' user to the group of the web server user (see Script below).

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

(SUSE=www, Red Hat/CentOS/Fedora=apache, Debian/Ubuntu=www-data)

Сценарий: Добавление нового пользователя 'otrs' и добавление его в группу.

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

```
linux:/opt# cd otrs/Kernel/
linux:/opt/otrs/Kernel# cp Config.pm.dist Config.pm
linux:/opt/otrs/Kernel# cd Config
linux:/opt/otrs/Kernel/Config# cp GenericAgent.pm.dist GenericAgent.pm
```

Сценарий: Копирование демонстрационных копий конфигурационных файлов

На завершающем этапе установки OTRS необходимо установить соответствующие права доступа к файлам. Для этого используйте сценарий **otrs.SetPermissions.pl**, находящийся в директории bin домашнего каталога пользователя 'otrs'. Скрипт можно вызвать со следующими параметрами:

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

Если ваш веб-сервер работает с теми же правами что и пользователь 'otrs', то команда установки надлежащих прав доступа будет выглядеть так: **otrs.SetPermissions.pl /opt/otrs --otrs-user=otrs --web-user=otrs**. На SUSE-системах веб-сервер работает с правами пользователя 'wwwrun'. На Debian-системах - 'www-data'. Для установки надлежащих прав доступа используйте команду **otrs.SetPermissions.pl /opt/otrs --otrs-user=otrs --web-user=wwwrun --otrs-group=nogroup --web-group=www**.

2.2. Установка Perl-модулей

OTRS requires some additional Perl modules, as described in Table 3-1. If you install OTRS from source, you will need to install these modules manually. This can be done either with the package manager of your Linux distribution (yast, apt-get) or, as 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. We recommend using your package manager if possible.

You can verify which modules you need to install with **otrs.CheckModules.pl**. This script is located in the bin directory, in the home directory of the 'otrs' user (see Script below). Please note that some modules are optional.

```
linux:~# cd /opt/otrs/bin/
linux:/opt/otrs/bin# ./otrs.CheckModules.pl
   o CGI.............................ok (v3.60)
   o Crypt::PasswdMD5.................ok (v1.3)
   o Crypt::SSLeay....................Not installed! (Optional - Required for Generic Interface
 SOAP SSL connections.)
   o CSS::Minifier....................ok (v0.01)
   o Date::Format.....................ok (v2.22)
   o Date::Pcalc......................ok (v1.2)

   ...
```

Сценарий: Проверка необходимых модулей.

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

Для установки любого из выше описанных модулей через CPAN, нужно выполнить команду **perl -e shell -MCPAN**. Оболочка Perl запустится в интерактивном режиме и модуль CPAN будет загружен. Если CPAN уже сконфигурирован, то с помощью команды **install** "имя модуля" (install module_name) можно начинать установку необходимых модулей. CPAN автоматически отслеживает зависимости между Perl-модулями и тут же оповещает, какие еще модули нужно установить.

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

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

Сценарий: Проверить синтаксис.

2.3. Настройка веб-сервера Apache

First of all, you should install the Apache2 web server and mod_perl; you'd typically do this from your systems package manager. Below you'll find the commands needed to set up Apache on the most popular Linux distributions.

```
    # rhel / centos:
    linux:# yum install httpd mod_perl

    # suse:
    linux:# zypper install apache2-mod_perl

    # debian/ubuntu:
    linux:# apt-get install apache2 libapache-mod-perl2
```

To access the web interface of OTRS via a short URL, Alias and ScriptAlias entries are needed. Most Apache installations have a `conf.d` directory included. On Linux systems you can usually find this directory under /etc/apache or /etc/apache2. Log in as root, change to the `conf.d` directory and copy the appropriate template in `/opt/otrs/scripts/apache2-httpd.include.conf` to a file called `otrs.conf` in the Apache configuration directory.

Перезагрузите веб-сервер, чтобы новые настройки вступили в силу. На большинстве систем запустить/перезагрузить веб-сервер можно с помощью команды **/etc/init.d/apache2 restart** (см. Сценарий ниже).

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

Сценарий: Перезапуск веб-сервера.

Теперь веб-сервер полностью настроен для работы OTRS.

2.4. Настройка базы данных

2.4.1. Установка базы данных OTRS вручную.

The recommended way to configure the database is to run the Web Installer. If you can't use this for some reason, you can also configure the database manually, as described in this chapter. If you can't use the web installer to setup the OTRS database, you have to set it up manually. Scripts with the SQL statements to create and configure the database are located in `scripts/database`, in the home directory of the 'otrs' user (see Script below).

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

Сценарий: Файлы, необходимые для создания и настройки базы данных.

При установке базы данных, для различных СУБД существует свой, определенный порядок обработки .sql-файлов.

Создание базы данных OTRS вручную, шаг за шагом

1. Создание базы данных: Используя интерфейс базы данных или свой любимый менеджер баз даных, создайте базу данных, которую планируете использовать для OTRS.

2. Создание таблиц: Используя файлы otrs-schema.DatabaseType.sql (например `otrs-schema.oracle.sql`, `otrs-schema.postgresql.sql`) можно создать таблицы в базе данных для OTRS.

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 database that you are using, you will need to use one of the following files: `otrs-initial_insert.mysql.sql`, `otrs-initial_insert.oracle.sql`, `otrs-initial_insert.postgresql.sql` or `otrs-initial_insert.mssql.sql` .

4. Создание связей между таблицами: Последний шаг - создание связей между различными таблицами базы данных OTRS Для этого используйте файл otrs-schema-post.DatabaseType.sql (например `otrs-schema-oracle.post.sql`, `otrs-schema-post.postgresql.sql`).

После завершения установки базы данных необходимо проверить и установить соответствующие права доступа для базы данных OTRS. Сделать это необходимо так, чтобы только один пользователь имел соответствующие права доступа. Настройка прав доступа отличается в зависимости от выбраного вами сервера базы данных и должна производится с помощью графического интерфейса базы данных или с помощью программы клиента.

If your database and the access rights are configured properly, you have to tell OTRS which database back-end you want to use and how the ticket system can connect to the database. Open the file `Kernel/Config.pm` located in the home directory of the 'otrs' user, and change the parameters shown in the script below according 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';
```

Сценарий: Настраиваемые параметры

2.5. Настрой планировщика задач (cron jobs) для OTRS

Для правильной работы системы OTRS необходим планировщик задач (cron jobs). Планировщик задач (cron jobs) должен запускаться с теми же правами, что и модули OTRS. Именно по этому cron jobs должен быть внесен в crontab-файл пользователя 'otrs'.

All scripts with the cron jobs are located in `var/cron`, in the home directory of the 'otrs' user (see Script below).

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

Сценарий: Файлы, необходимые для создания заданий планировщика задач (cron jobs).

These scripts have a suffix of '.dist'. You should copy them to files with the suffix removed. If you use bash, you might want to use the command listed in Script below.

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

```
linux:/opt/otrs/var/cron#
```

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

В Таблице 3-2 приводятся различные задания cron jobs.

Таблица 2.1. Описание некоторых сценариев планировщика задач cron job.

Сценарий	Назначение
aaa_base	Определяет основные настройки для crontab пользователя 'otrs'.
cache	Удаляет из диска устаревшие ("просроченные") кэш-записи. Очищает кэш-погрузчик для CSS и JavaScript файлов.
fetchmail	Этот сценарий может быть использован в том случае, если новые емейлы будут поступать в систему обработки заявок через fetchmail.
generic_agent	Выполняет задания из GenericAgent, которые не сохраняются в базе данных, а в собственных конфиг-файлах.
generic_agent-database	Выполняет задания из GenericAgent, которые хранятся в базе данных.
pending_jobs	Проверяет систему на наличие заявок, ожидающих решения, закрывает их или отсылает напоминание, если это необходимо.
postmaster	Проверяет очередь сообщений системы обработки заявок, и доставляет те сообщения, которые находятся в этой очереди.
postmaster_mailbox	Получает почту с POP3-счетов, которые были указаны в админке, в разделе "Учетные записи PostMaster".
rebuild_ticket_index	Восстанавливает индекс заявки, что значительно повышает скорость просмотра заявок в разделе QueueView.
session	Удаляет старые и больше не используемые ID-сессий (session IDs).
unlock	Открывает заявки, которые были ранее заблокированы в системе.

To setup all cron jobs, the script `bin/Cron.sh` located in the home directory of the 'otrs' user can be used. When this script is executed, it needs a parameter to specify whether you want to install, remove, or reinstall the cron jobs. The following parameters can be used:

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

Because the cron jobs need to be installed in the crontab file of the 'otrs' user, you need to be logged in as 'otrs'. If you are logged in as root, you can switch to 'otrs' with the command **su otrs**. Execute the commands specified in Script below to install the cron jobs.

Предупреждение

Обратите внимание, что при использовании файла `Cron.sh` другие задания, установленные в crontab-файле для пользователя 'otrs' будут перезаписаны или удалены. Внесите все необходимые изменения в файл `Cron.sh`, чтобы сохранить другие crontab-задания.

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

Сценарий: Установка cron jobs.

Команда **crontab -l -u otrs**, которую можно запускать с правами администратора, отображает содержание crontab-файла пользователя 'otrs', так что можно легко проверить правильность расположения всех записей (см. Сценарий ниже).

```
linux:/opt/otrs/bin# crontab -l -u otrs
# --
# cron/aaa_base - base crontab package
# Copyright (C) 2001-2014 OTRS AG, http://otrs.com/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# Who gets the cron emails?
MAILTO="root@localhost"

# --
# cron/cache - delete expired cache
# Copyright (C) 2001-2014 OTRS AG, http://otrs.com/
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# delete expired cache weekly (Sunday mornings)
20 0 * * 0  $HOME/bin/otrs.CacheDelete.pl --expired >> /dev/null
30 0 * * 0  $HOME/bin/otrs.LoaderCache.pl -o delete >> /dev/null

# --
# cron/fetchmail - fetchmail cron of the OTRS
# Copyright (C) 2001-2014 OTRS AG, http://otrs.com/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# fetch every 5 minutes emails via fetchmail
#*/5 * * * * /usr/bin/fetchmail -a >> /dev/null

# --
# cron/generic_agent - otrs.GenericAgent.pl cron of the OTRS
# Copyright (C) 2001-2014 OTRS AG, http://otrs.com/
# --
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# start generic agent every 20 minutes
*/20 * * * * $HOME/bin/GenericAgent.pl >> /dev/null
# example to execute GenericAgent.pl on 23:00 with
# Kernel::Config::GenericAgentMove job file
```

```
#0 23 * * * $HOME/bin/otrs.GenericAgent.pl -c "Kernel::Config::GenericAgentMove" >> /dev/null
# --
# cron/generic_agent - GenericAgent.pl cron of the OTRS
# Copyright (C) 2001-2014 OTRS AG, http://otrs.com/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# start generic agent every 10 minutes
*/10 * * * * $HOME/bin/otrs.GenericAgent.pl -c db >> /dev/null
# --
# cron/pending_jobs - pending_jobs cron of the OTRS
# Copyright (C) 2001-2014 OTRS AG, http://otrs.com/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# check every 120 min the pending jobs
45 */2 * * * $HOME/bin/otrs.PendingJobs.pl >> /dev/null
# --
# cron/postmaster - postmaster cron of the OTRS
# Copyright (C) 2001-2014 OTRS AG, http://otrs.com/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# check daily the spool directory of OTRS
10 0 * * *  $HOME/bin/otrs.ReprocessMails.pl >> /dev/null
# --
# cron/postmaster_mailbox - postmaster_mailbox cron of the OTRS
# Copyright (C) 2001-2014 OTRS AG, http://otrs.com/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# fetch emails every 10 minutes
*/10 * * * * $HOME/bin/otrs.PostMasterMailbox.pl >> /dev/null
# --
# cron/rebuild_ticket_index - rebuild ticket index for OTRS
# Copyright (C) 2001-2014 OTRS AG, http://otrs.com/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# just every day
01 01 * * * $HOME/bin/otrs.RebuildTicketIndex.pl >> /dev/null

# --
# cron/session - delete old session ids of the OTRS
# Copyright (C) 2001-2014 OTRS AG, http://otrs.com/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# delete every 120 minutes old/idle session ids
55 */2 * * * $HOME/bin/otrs.DeleteSessionIDs.pl --expired >> /dev/null

# --
# cron/unlock - unlock old locked ticket of the OTRS
# Copyright (C) 2001-2014 OTRS AG, http://otrs.com/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# unlock every hour old locked tickets
35 * * * * $HOME/bin/otrs.UnlockTickets.pl --timeout >> /dev/null

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

Сценарий: Файл Crontab.

2.6. OTRS Scheduler Service

OTRS comes with a scheduler service that is used to perform asynchronous tasks.

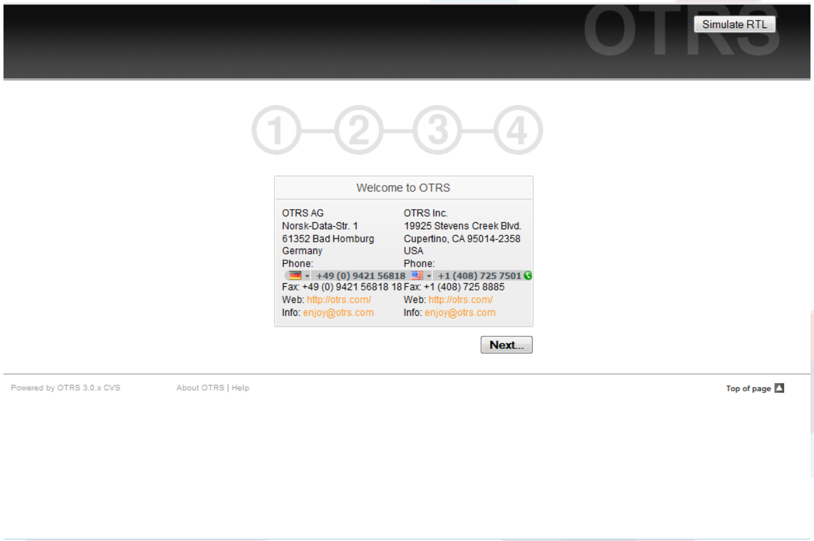
The OTRS RPMs will set up the Scheduler Service automatically. If you install from source, you can install the service by copying the `scripts/otrs-scheduler-linux` file to `/etc/init.d` and giving it the appropriate permissions. This will make sure the scheduler service starts when the system starts up.

3. Using the web installer

You can use the OTRS Web Installer, after you installed the OTRS software, to set up and configure the OTRS database. The Web Installer is a web page you can visit in your browser. The URL for the web installer is http://localhost/otrs/installer.pl .

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

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

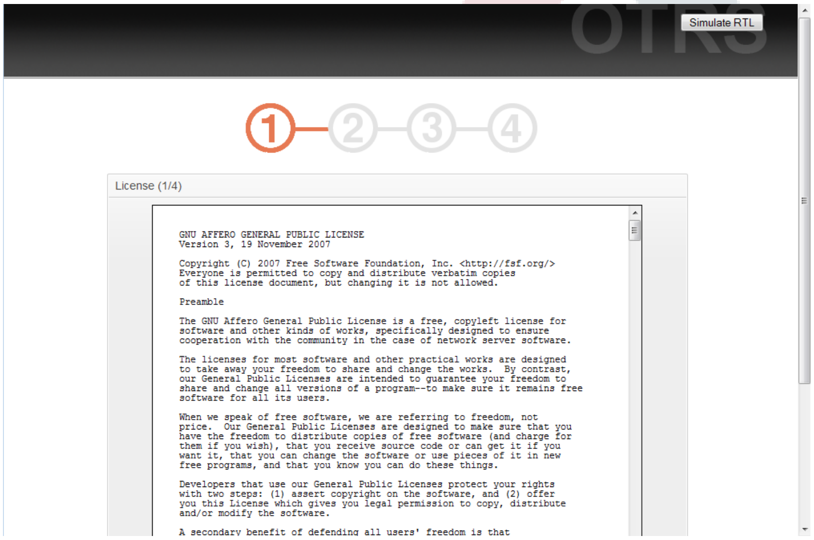


Welcome to OTRS

OTRS AG	OTRS Inc.
Norsk-Data-Str. 1	19925 Stevens Creek Blvd.
61352 Bad Homburg	Cupertino, CA 95014-2358
Germany	USA
Phone:	Phone:
+49 (0) 9421 56818	+1 (408) 725 7501
Fax: +49 (0) 9421 56818 18	Fax: +1 (408) 725 8885
Web: http://otrs.com/	Web: http://otrs.com/
Info: enjoy@otrs.com	Info: enjoy@otrs.com

Рисунок: Экран приветствия.

2. Прочитайте соглашения лицензии GNU Affero General Public License (см. Рисунок ниже) и примите их, нажав на соответствующую кнопку внизу страницы.

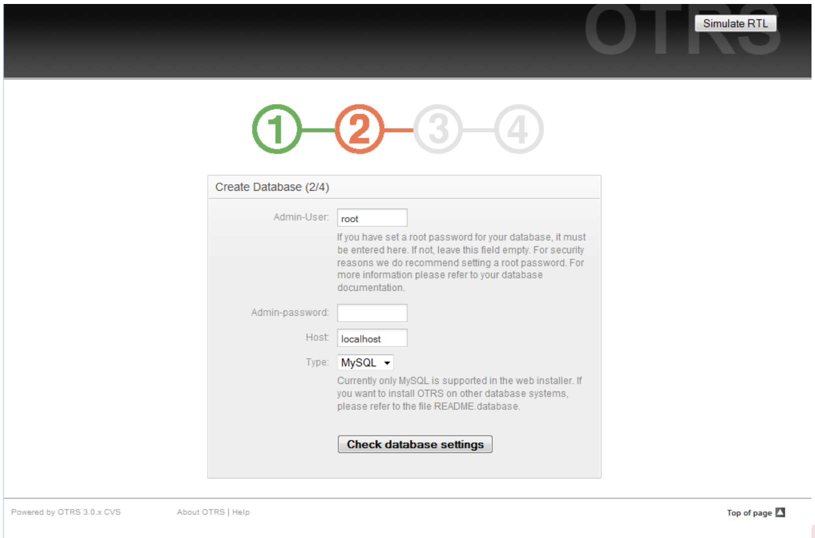


License (1/4)

```
GNU AFFERO GENERAL PUBLIC LICENSE
Version 3, 19 November 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The GNU Affero General Public License is a free, copyleft license for
software and other kinds of works, specifically designed to ensure
cooperation with the community in the case of network server software.

The licenses for most software and other practical works are designed
to take away your freedom to share and change the works.  By contrast,
our General Public Licenses are intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.

When we speak of free software, we are referring to freedom, not
price.  Our General Public Licenses are designed to make sure that you
have the freedom to distribute copies of free software (and charge for
them if you wish), that you receive source code or can get it if you
want it, that you can change the software or use pieces of it in new
free programs, and that you know you can do these things.

Developers that use our General Public Licenses protect your rights
with two steps: (1) assert copyright on the software, and (2) offer
you this License which gives you legal permission to copy, distribute
and/or modify the software.

A secondary benefit of defending all users' freedom is that
```

Рисунок: GNU Affero General Public License.

3. Choose the database that you want to use with OTRS. If you choose MySQL, PostgreSQL or Microsoft SQL Server as a database, you can also select here if you want the web installer to create a database for you or if your database administrator has already created an empty database for you that you would like to use. After that, click the Next button (see figure below).



Create Database (2/4)

Admin-User: root

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

Admin-password:

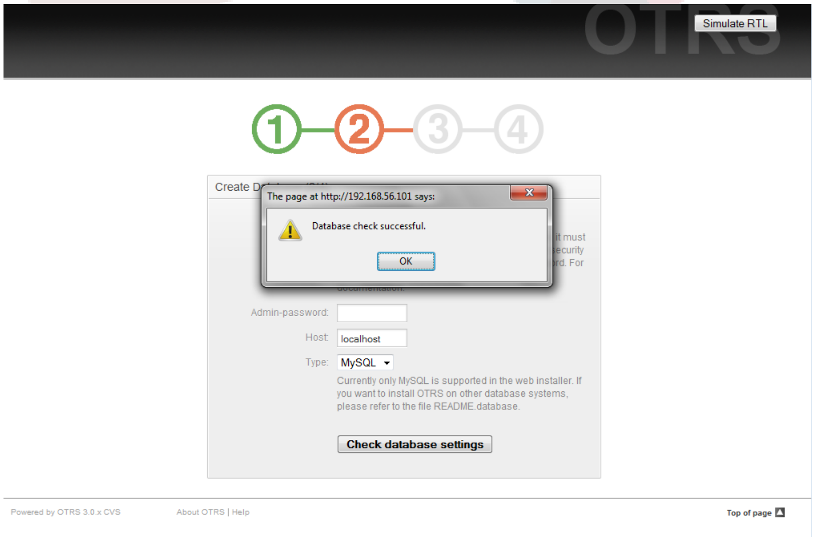
Host: localhost

Type: MySQL

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

Figure: Database Selection.

4. Depending of the database you chose and if you wanted the web installer to create a database or use an existing one in the previous step, this screen might differ a little. Enter the credentials for the database in this screen.



The page at http://192.168.56.101 says:

Database check successful.

Admin-password:

Host: localhost

Type: MySQL

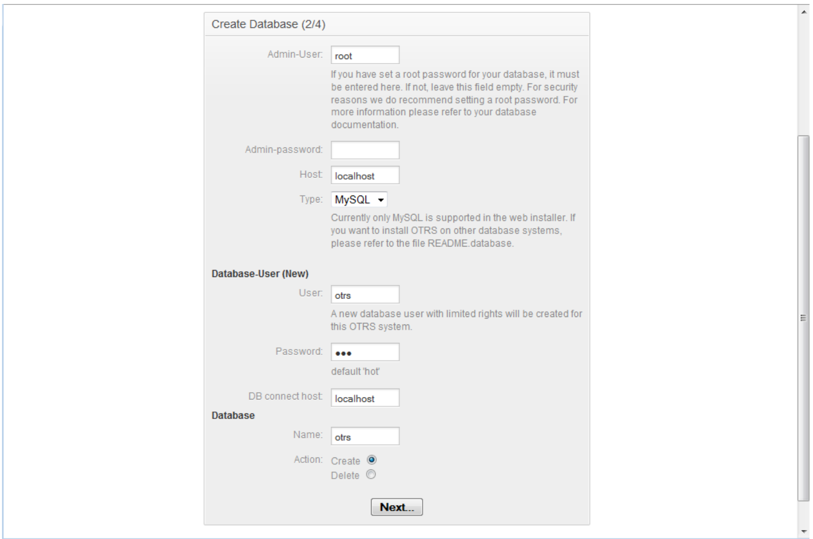
Currently only MySQL is supported in the web installer. If you want to install OTRS on other database systems, please refer to the file README.database.

Figure: Database credentials.

5. Create a new database user, choose a name for the database and click on 'Next' (see Figure below).

Предупреждение

OTRS will generate a strong password for you. It's possible to enter your own password if you prefer this. The password will be written to the configuration file `Kernel/Config.pm` so there is no need to remember this password.



Create Database (2/4)

Admin-User: root

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

Admin-password:

Host: localhost

Type: MySQL

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

Database-User (New)

User: otrs

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

Password: •••

default 'hot'

DB connect host: localhost

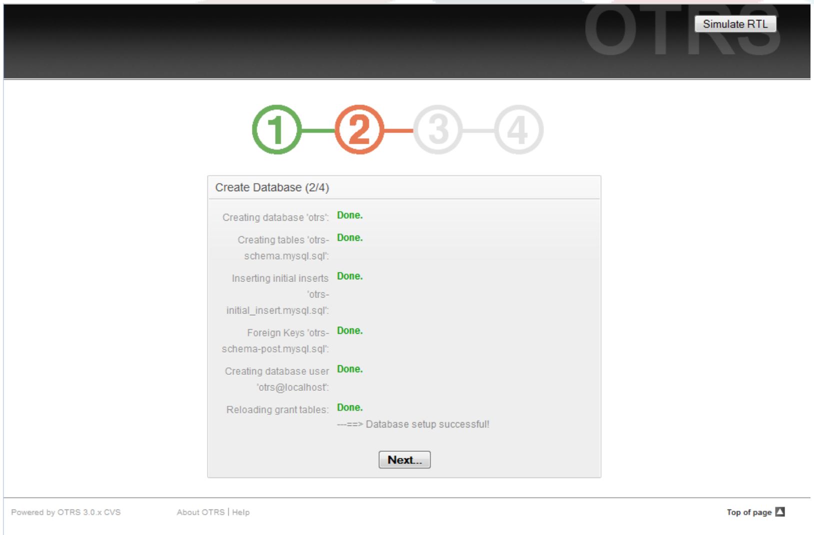
Database

Name: otrs

Action: Create ● Delete ○

Рисунок: Настройки базы данных.

6. The database will be created if needed, and populated, as shown in this image. Click 'Next' to go to the next screen.



Create Database (2/4)

Creating database 'otrs': Done.

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

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

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

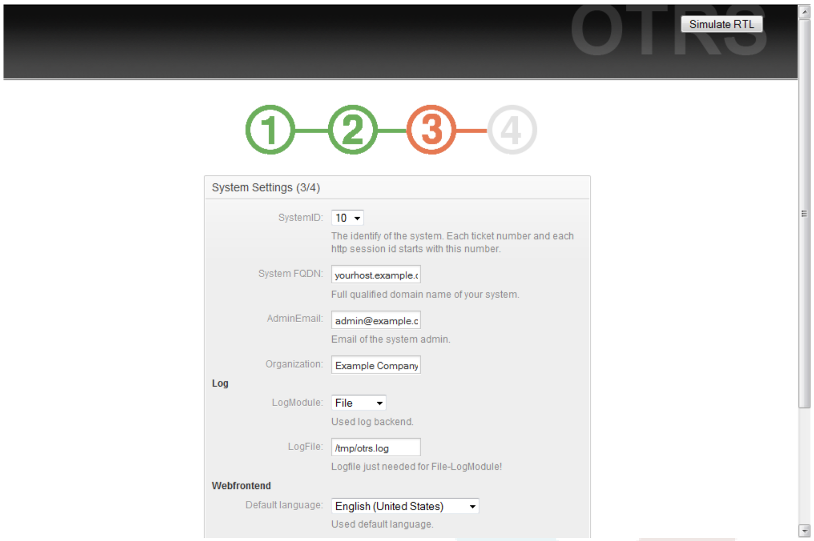
Creating database user 'otrs@localhost': Done.

Reloading grant tables: Done.

---==> Database setup successful!

Figure: Successful database setup.

7. Provide all the required system settings and click on 'Next' (see Figure below).



System Settings (3/4)

SystemID: 10
The identify of the system. Each ticket number and each http session id starts with this number.

System FQDN: yourhost.example.c
Full qualified domain name of your system.

AdminEmail: admin@example.c
Email of the system admin.

Organization: Example Company

Log

LogModule: File
Used log backend.

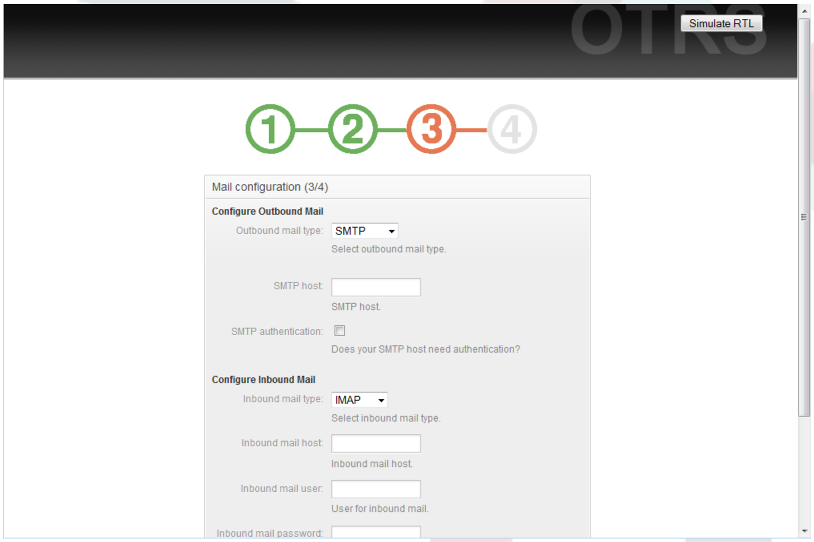
LogFile: /tmp/otrs.log
Logfile just needed for File-LogModule!

Webfrontend

Default language: English (United States)
Used default language.

Рисунок: Настройки системы.

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



Mail configuration (3/4)

Configure Outbound Mail

Outbound mail type: SMTP
Select outbound mail type.

SMTP host:
SMTP host.

SMTP authentication: □
Does your SMTP host need authentication?

Configure Inbound Mail

Inbound mail type: IMAP
Select inbound mail type.

Inbound mail host:
Inbound mail host.

Inbound mail user:
User for inbound mail.

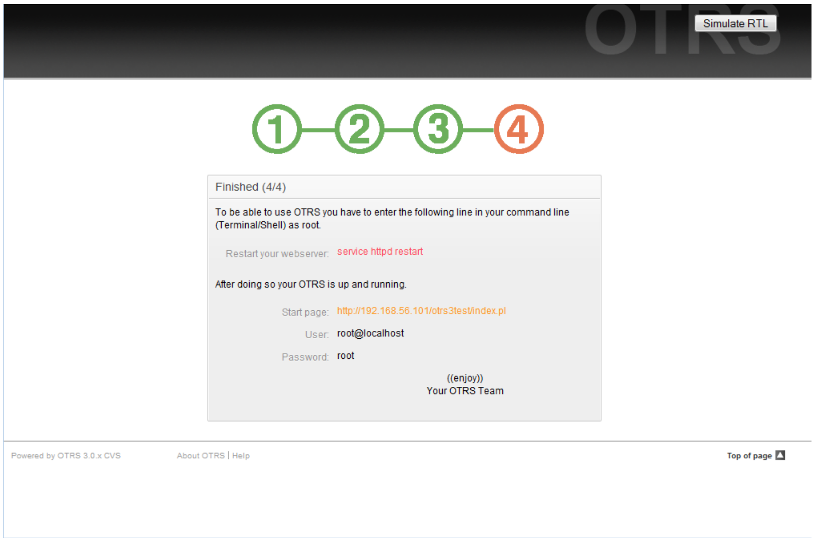
Inbound mail password:

Рисунок: Настройка электронной почты.

9. Congratulations! Now the installation of OTRS is finished and you should be able to work with the system (see Figure below). To log into the web interface of OTRS, use the address http://localhost/otrs/index.pl from your web browser. Log in as OTRS administrator, using the username 'root@localhost' and the generated password. After that, you can configure the system to meet your needs.

Предупреждение

Please write down the generated password for the 'root@localhost' account.



Finished (4/4)

To be able to use OTRS you have to enter the following line in your command line (Terminal/Shell) as root.

Restart your webserver: service httpd restart

After doing so your OTRS is up and running.

Start page: http://192.168.56.101/otrs3test/index.pl

User: root@localhost

Password: root

((enjoy))
Your OTRS Team

Figure: Web installer final screen.

4. Обновление OTRS-фреймворка

These instructions are for people upgrading OTRS from version *3.2* to *3.3*, and apply both for RPM and source code (tarball) upgrades.

Примечание

Please note that OTRS 3.3 requires at least perl version 5.10.0. Make sure before you plan your upgrade that your server runs this version. You can check the version with the command **perl -v** on the command line.The only known Linux distribution that uses perl 5.8 and is still supported by its vendor is Red Hat Enterprise Linux (RHEL) 5 and its community supported derivative CentOS 5. If you're on any of these platforms and you plan to upgrade to OTRS 3.3 you should also plan migrating your operating system to a version with a supported perl, such as RHEL 6 or CentOS 6.

If you are running a lower version of OTRS you have to follow the upgrade path to 3.1 first (1.1->1.2->1.3->2.0->2.1->2.2->2.3->2.4->3.0->3.1->3.2->3.3 ...)! You need to perform a full upgrade to every version in between, including database changes and the upgrading perl script.

Please note that if you upgrade from OTRS 2.2 or earlier, you must take an extra step; please read http://bugs.otrs.org/show_bug.cgi?id=6798.

Within a single minor version you can skip patch level releases if you want to upgrade. For instance you can upgrade directly from OTRS 3.3.1 to version 3.3.4. If you need to do such a "patch level upgrade", you should skip steps 9 and 13-16.

Please note that if you perform a patch level upgrade from OTRS 3.3.6 or earlier, you need to activate a new cron job that controls the scheduler process. Please copy the file `/opt/otrs/var/cron/scheduler_watchdog.dist` to `/opt/otrs/var/cron/scheduler_watchdog` and update your crontab with `/opt/otrs/bin/Cron.sh start` (as "otrs" user, not root).

1. Остановите все соответствующие службы.

   в т.ч. (зависящие от использующихся услуг):

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

2. Сделайте резервные копии всех данных из $OTRS_HOME (по умолчанию OTRS_HOME=/opt/otrs):

   - `Kernel/Config.pm`
   - `Kernel/Config/GenericAgent.pm`
   - `Kernel/Config/Files/ZZZAuto.pm`
   - `var/*`
   - а также базу данных

3. Убедитесь, что вы создали резервную копию всех данных ;-)

4. Установить новую систему (опционально)

   Если есть возможность, сначала попробуйте произвести установку на отдельном тестовом компьютере.

5. Установка нового релиза (из tar-архива или с помощью RPM-пакетов).

   - С помощью тарбола (tarball):

     ```
     shell> cd /opt
     shell> mv otrs otrs-old
     shell> tar -xzf otrs-x.x.x.tar.gz
     shell> mv otrs-x.x.x otrs
     ```

     Restore old configuration files

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

     Restore TicketCounter.log

     In order to let OTRS continue with the correct ticket number, restore the `TicketCounter.log` to `$OTRS_HOME/var/log/` (default: `OTRS_HOME=/opt/otrs`). This is especially important if you use incremental ticketnumbers.

     Restore article data

     If you configured OTRS to store article data in the filesystem you have to restore the `article` folder to `$OTRS_HOME/var/` (default: `OTRS_HOME=/opt/otrs`).

   - С помощью RPM-пакетов:

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

В этом случае обновление из RPM автоматически восстанавливает старые конфигурационные файлы.

6. Собственные темы

   Note: The OTRS themes between 3.2 and 3.3 are NOT compatible, so don't use your old themes!

   Темы хранятся в $OTRS_HOME/Kernel/Output/HTML/*/*.dtl (default: OTRS_HOME=/opt/otrs).

   Please also note that dtl-Files must be in utf-8 from OTRS 3.3 on.

7. Установка прав доступа к файлам.

   В случае использования тарбола (tarball), выполните

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

   с правами, необходимыми для установки системы.

8. Check needed Perl modules

   Verify that all needed perl modules are installed on your system and install any modules that might be missing.

   ```
   shell> /opt/otrs/bin/otrs.CheckModules.pl
   ```

9. Схема обновления:

   Note: new tables created in the MySQL UPGRADING process will be created with the default table storage engine set in your MySQL server. In MySQL 5.5 the new default type is InnoDB. If existing tables, e.g. "users", have the table storage engine, e.g. MyISAM, then an error will be displayed when creating the foreign key constraints.

   You have two options: (1) you can change the default storage engine of MySQL back to MyISAM so that new tables will have the same engine as the existing tables, or (2) change the existing tables to use InnoDB as storage engine.

   Any problems with regards to the storage engine will be reported by the otrs.CheckDB.pl script, so please run it to check for possible issues.

   ```
   shell> cd /opt/otrs/

   # MySQL:
   shell> bin/otrs.CheckDB.pl
   shell> cat scripts/DBUpdate-to-3.3.mysql.sql | mysql -p -f -u root otrs

   # PostgreSQL 8.2+:
   shell> cat scripts/DBUpdate-to-3.3.postgresql.sql | psql --set ON_ERROR_STOP=on --single-
   transaction otrs otrs

   # PostgreSQL, older versions:
   ```

```
shell> cat scripts/DBUpdate-to-3.3.postgresql_before_8_2.sql | psql --set ON_ERROR_STOP=on --
single-transaction otrs otrs
```

Note: If you use PostgreSQL 8.1 or earlier, you need to activate the new legacy driver for these older versions. Do this by adding a new line to your `Kernel/Config.pm` like this:

```
$Self->{DatabasePostgresqlBefore82} = 1;
```

Запустите сценарий перемещения (как пользователь 'otrs', а НЕ root):

```
shell> scripts/DBUpdate-to-3.3.pl
```

Do not continue the upgrading process if this script does not work properly for you. Otherwise data loss may occur.

10. Refresh the configuration and delete caches. Please run (as user 'otrs', NOT as 'root'):

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

11. Перезапустите сервисы.

    в т.ч. (зависящие от использующихся услуг):

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

Теперь вы можете войти в систему.

12. Проверьте установленные пакеты

    Проверьте в менеджере пакетов все ли пакеты по прежнему помечены как правильно установлены или какие либо из них требуют переустановки или же обновления.

    The following packages are automatically uninstalled after the upgrade process (if they where installed before):

    - OTRSPostMasterFilterExtensions
    - OTRSFreeTextFromCustomerUser
    - OTRSExternalTicketNumberRecognition
    - OTRSDashboardQueueOverview
    - OTRSImportantArticles
    - OTRSImportantArticlesITSM
    - OTRSDashboardTicketCalendar
    - OTRSMultiServiceSelect

- OTRSMultiQueueSelect
- OTRSDynamicFieldMultiLevelSelection
- OTRSEventBasedTicketActions
- OTRSTicketAclEditor
- OTRSCustomerProcessSelection
- OTRSACLExtensions
- OTRSGenericStandardTemplates
- OTRSExtendedDynamicDateFieldSearch
- OTRSDashboardTicketOverviewFilters

13Check config settings of OTRSFreeTextFromCustomerUser

Note: This only applies if you used the package OTRSFreeTextFromCustomerUser previously.

If you used this module previously, you need to reconfigure it. The module is automatically uninstalled by the upgrading script as it is now a part of the OTRS framework.

If you want to keep using it, please enable the setting "Ticket::EventModulePost###930-DynamicFieldFromCustomerUser" to activate this feature and configure the mapping in the setting "DynamicFieldFromCustomerUser::Mapping".

14Import your ACLs to the new ACL editor (optional)

In OTRS 3.3, there is a graphical editor for ACLs in the administration interface. You will need to import your existing ACLs (e.g. in Config.pm or additional files) to the editor by using bin/otrs.ImportACLsFromConfig.pl in order to make them available in the editor. Please make sure to delete any ACLs from Config.pm (or other files) after successfully finishing the import procedure. Also you will need to use the deploy button in the ACL administration frontend in order to re-deploy the imported ACLs to your system.

15Update your web server configuration

Note: this applies only if you use the Apache web server, and do not use the configuration file directly from the OTRS installation directory (e. g. with a symlink from the Apache configuration directory).

Please update the the Apache configuration file for OTRS as there have been several changes (see `scripts/apache2-httpd.include.conf`).

16Update and activate cronjobs

There are several OTRS default cronjobs in $OTRS_HOME/var/cron/*.dist. They can be activated by copying them without the ".dist" filename extension. Do this to make sure you get the latest versions of the cronjobs and new cronjobs as well.

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

Please check the copied files and re-apply any customizations that you might have made.

To schedule these cronjobs on your system, you can use the script Cron.sh. Make sure to execute it as the OTRS system user!

```
shell> /opt/otrs/bin/Cron.sh start
```

17Молодцы!

5. Обновление с помощью Windows Installer

If you installed OTRS on Windows, it is possible to upgrade seamlessly by downloading the new Windows Installer and running it. The installer will automatically detect the OTRS installation and perform an upgrade on it. The installer will stop the OTRS services and start them after the installation is complete, so your users will experience downtime. Make sure you plan some time for the upgrade. Always make sure you have a backup of your server before you start the upgrade procedure!

5.1. Performing patch level upgrades on Windows

If you need to install a 'patch level' upgrade (i.e. from OTRS 3.3.1 to OTRS 3.3.2) on Windows all you need to do is to download and run the new installer. Of course you should make sure you have a backup in place before you start the upgrade!

5.2. Performing minor level upgrades on Windows

Minor upgrades (from OTRS 3.2 to OTRS 3.3) take a little more time as there is also a database upgrade step involved. The installer will detect this and will automatically perform the upgrade for you. After installation you should still perform steps 12, 13 and 14 of the 'regular' upgrading section.

Please note that if you have any additional packages such as Feature Add-Ons installed on your OTRS system, you'll need to upgrade these as well. Make sure you obtain these before you start the upgrade. After upgrade is complete you can upgrade these packages via the Package Installer Web GUI. Packages from the 'standard' repositories such as Survey, FAQ and ITSM can be upgraded from the Package Manager after the framework upgrade is complete.

6. Дополнительные приложения

Для разширения функциональности OTRS-фреймворка, можно также установить дополнительные пакеты приложений. Это можно сделать с помощью менеджера пакетов из Панели Администрирования, который загружает приложения из онлайн репозитория и управляет зависимостями между этими пакетами. Также приложения можно установить из локальных файлов.

6.1. FAQ (Часто Задаваемые Вопросы)

The FAQ is the Knowledge Base component. It supports editing and viewing of FAQ articles. Articles can be viewed and restricted to agents, customer users, or anonymous users. These can also be structured into groups, and be read in different languages.

Глава 3. Первые шаги

The goal of this chapter is to provide a brief overview of OTRS and the structure of its web interface. The terms 'agents', 'customers', and 'administrators' are introduced. We also login as the OTRS admininstrator and take a closer look at the user preferences available on every account.

1. Веб-интерфейс для агентов

Агентский веб-интерфейс позволяет агентам отвечать на вопросы клиентов, создавать новые заявки для клиентов и агентов, создавать заявки на основе телефонных звонков клиентов, писать и редактировать записи FAQ-модуля, редактировать данные клиентов и так далее.

Предположим, что OTRS доступен по ссылке *http://www.example.com*, таким образом, агенты и OTRS-администраторы могут войти в OTRS набрав в строке браузера следующий URL-адрес: http://www.example.com/otrs/index.pl (см. Рисунок ниже).

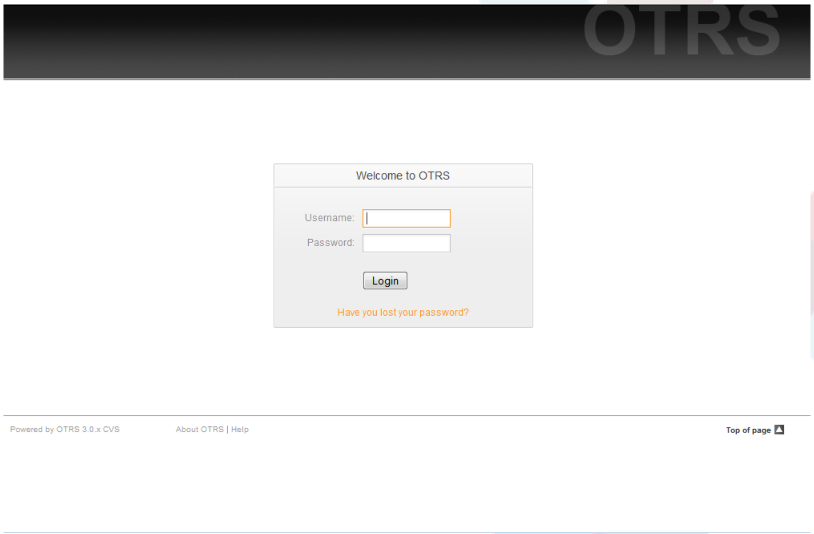


Рисунок: Агентский интерфейс, экран входа.

2. Пользовательский веб-интерфейс

Customers have a separate web interface in OTRS through which they can create new accounts, change their account settings, create and edit tickets, get an overview on tickets that they have created, etc.

Continuing the above example, the customer login screen can be reached by using the URL http://www.example.com/otrs/customer.pl with a web browser (see Figure below).

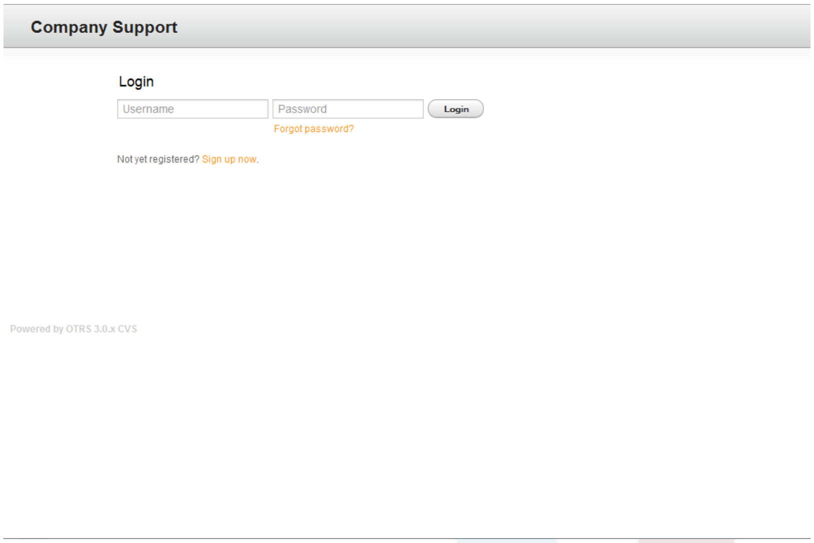
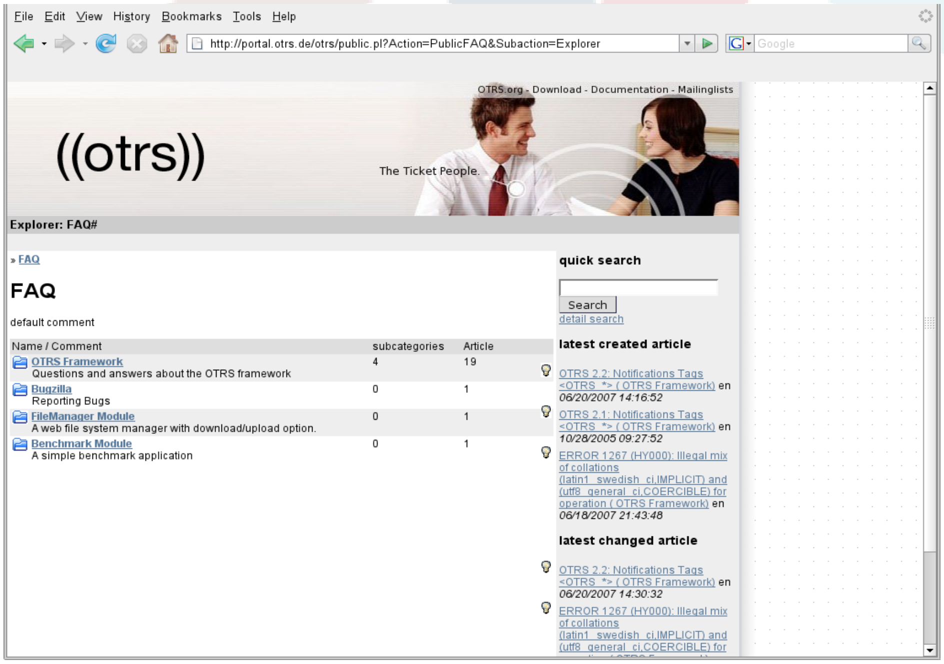


Рисунок: Пользовательский веб-интерфейс, страница входа.

3. Общедоступный веб-интерфейс

In addition to the web interfaces for agents and customers, OTRS also has a public web interface which is available through the FAQ-Module. This module needs to be installed separately. It provides public access to the FAQ system and lets visitors search through FAQ entries without any special authorization.

In our example, the public web interface can be reached via either of the following URLs: http://www.example.com/otrs/faq.pl , http://www.example.com/otrs/public.pl


File Edit View History Bookmarks Tools Help
http://portal.otrs.de/otrs/public.pl?Action=PublicFAQ&Subaction=Explorer
Google
OTRS.org - Download - Documentation - Mailinglists
((otrs))
The Ticket People.
Explorer: FAQ#
» FAQ
quick search
Search
detail search

FAQ

default comment

Name / Comment	subcategories	Article
OTRS Framework Questions and answers about the OTRS framework	4	19
Bugzilla Reporting Bugs	0	1
FileManager Module A web file system manager with download/upload option.	0	1
Benchmark Module A simple benchmark application	0	1

latest created article

OTRS 2.2: Notifications Tags <OTRS_*> (OTRS Framework) en 06/20/2007 14:16:52

OTRS 2.1: Notifications Tags <OTRS_*> (OTRS Framework) en 10/28/2005 09:27:52

ERROR 1267 (HY000): Illegal mix of collations (latin1_swedish_ci,IMPLICIT) and (utf8_general_ci,COERCIBLE) for operation (OTRS Framework) en 06/18/2007 21:43:48

latest changed article

OTRS 2.2: Notifications Tags <OTRS_*> (OTRS Framework) en 06/20/2007 14:30:32

ERROR 1267 (HY000): Illegal mix of collations (latin1_swedish_ci,IMPLICIT) and (utf8_general_ci,COERCIBLE) for

Рисунок: Общий веб-интерфейс.

4. Первое применение

Access the login screen as described in the section Agent web interface . Enter a user name and password. Since the system has just been installed and no users have yet been created, login as OTRS administrator first, using 'root@localhost' for username and 'root' for password.

Предупреждение

Данные доступа этого аккаунта действительны для каждой копии новоустановленной системы OTRS. Именно по этой причине пароль OTRS-администратора необходимо изменить как можно быстрее. Это можно сделать после входа в систему с правами администратора перейдя по ссылке preferences.

Если не хотите входить в систему с правами OTRS-администратора, просто введите имя и пароль своего обычного агентского аккаунта.

In case you have forgotten your password, you can request the system for a new password. Simply press the link below the Login button, enter the mail address that is registered for your OTRS account into the input field, and press the Submit button (see Figure).

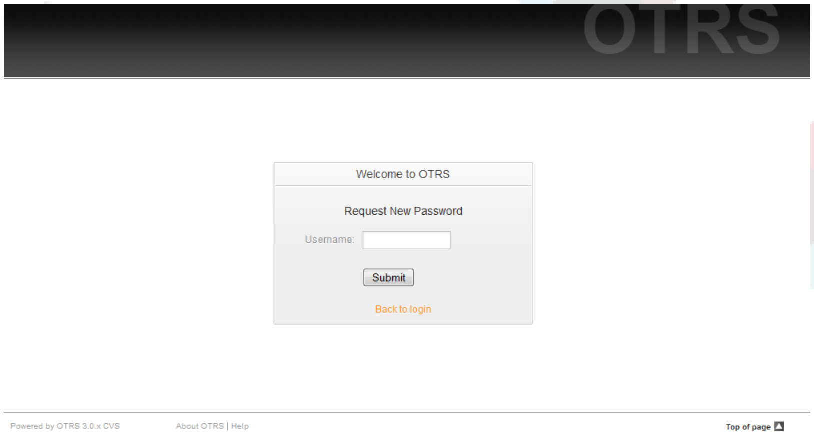
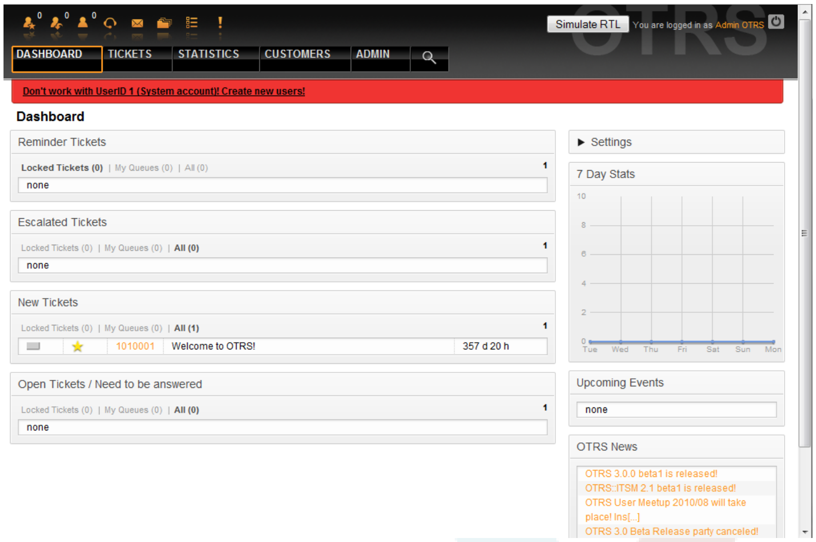


Рисунок: Запрос на восстановление пароля.

5. Веб-интерфейс - обзор

Upon successfully logging into the system, you are presented with the Dashboard page (see Figure below). It shows your locked tickets, allows direct access through menus to the queue, status and escalation views, and also holds options for creation of new phone and e-mail tickets. It also presents a quick summary of the tickets using different criteria.



You are logged in as Admin OTRS

Don't work with UserID 1 (System account)! Create new users!

Dashboard

Reminder Tickets

Locked Tickets (0) | My Queues (0) | All (0) 1

none

Escalated Tickets

Locked Tickets (0) | My Queues (0) | All (0) 1

none

New Tickets

Locked Tickets (0) | My Queues (0) | All (1) 1

	★	1010001	Welcome to OTRS!	357 d 20 h

Open Tickets / Need to be answered

Locked Tickets (0) | My Queues (0) | All (0) 1

none

7 Day Stats

Upcoming Events

none

OTRS News

OTRS 3.0.0 beta1 is released!
OTRS::ITSM 2.1 beta1 is released!
OTRS User Meetup 2010/08 will take place! Ins[...]
OTRS 3.0 Beta Release party canceled!

Рисунок: Панель управления веб-интерфейса агентов

To improve clarity, the general web interface is separated into different areas. The top row of each page shows some general information such the current username, the logout button, icons listing the number of locked tickets with direct access to them, links to create a new phone/e-mail ticket, etc. There are also icons to go to the queue, status, and escalation views.

Below the icons row is the navigation bar. It shows a menu that enables you to navigate to different areas or modules of the system, letting you execute some global actions. Clicking on the Dashboard button takes you to the dashboard. If you click on the Tickets button, you will get a submenu with options to change the ticket's view, create a new ticket (phone/e-mail) or search for a specific ticket. The Statistics button presents a menu that allows you to choose from an overview of the registered statistics, creating a new one or importing an existing one. The Customers button leads you to the Customer Management screen. By clicking the Admin button, you can access all of the administrator modules, which allows you to create new agents, queues, etc. There is also a Search button to make ticket searches.

If any associated applications are also installed, e.g. the FAQ or the Survey, buttons to reach these applications are also displayed.

Красная полоса, которая находится под панелью навигации предназначена для отображения различных системных сообщений. Если войти в систему с правами OTRS-администратора, будет выведено сообщение о том, что в системе не следует работать с правами администратора.

Below the title of the section you are currently in, there are several subsections containing relevant information about the screen you are working on, each one in a separate box.

These boxes contains the main part of each screen, usually they are displayed in one or several columns, each box can store relevant information about the current screen like for example instructions, advises, overviews, etc. Also is displayed the form or tool necessary for performing the action associated to each screen, like for example, add, update or delete records, check the log, change configuration settings, etc.

Finally at the bottom of the page, the site footer is displayed (see Figure below). It contains links to directly access the OTRS official web site, or go to the Top of the page.

Normally the icon row, navigation bar and footer are the same in all the screens over the web interface.

Рисунок: Нижний колонтитул.

6. The Dashboard

The Dashboard is the main page of the system, here you can get an overview about the tickets and other stuff related to the ticket activity. It's thinking to be the starting point for the daily work of an agent, by default it presents a quick summary of the tickets which are pending, escalated, new, and open, among other information.

One of the most important features about Dashboard is that is completely customizable. That means you that can configure each part as you want, showing or hiding elements. It's even possible to relocate this elements within the same column by clicking on and dragging the element's header, and dropping them elsewhere. Each element is named "Widget", the system has some widgets ready to use out of the box, but the modular design of the dashboard screen is prepared to integrate custom widgets easily.

The content of this screen is arranged in two main columns, on the left column you normally can see information about tickets classified by their states like: reminder, escalated, new, and open. On each of this widgets you can filter the results to see all of the tickets that you are allowed to access, tickets you have locked, the ones that are located in agent defiend queues, among other filters. There are also other kind of widgets in this column and they are all described below.

Left column dashboard widgets.

- Ticket List Widgets

  Widgets under this category share same overall behavior, look and feel. This widgets shows a list of tickets on a determined state. the amount of tickets display on each list page can be configured in widget options (they appear when you hover the mouse pointer over the top right part of the widget). This widgets support the following filters:

  - My locked tickets

    The tickets that the logged agent has locked.

  - My watched tickets

    The tickets that the logged agent has in his/her watched list, requires Ticket::Watcher setting to be turned on to be displayed.

  - My responsibilities

    The tickets that the logged agent is set as responsible, Ticket::Responsible setting is required to be turned on in order to make this filter visible.

  - Tickets in My Queues

    The tickets that are on queues where the agent define as "My Queues".

- All Tickets

  All the tickets where the agent has access.

  This widgets are:

  - Reminder Tickets

    Tickets that are set as pending and the reminder date has been reach.

  - Escalated Tickets

    Tickets that are escalated.

  - New Tickets

    Tickets that have state "New".

  - Open Tickets / Need to be answered

    Tickets that have state "Open" and are ready for work with them.

- Events Ticket Calendar

  A calendar event (for this widget) is defined when a new ticket is created, the Events Ticket Calendar feature has to be enabled, and it requires two new fields to be displayed in ticket creation screens, one for the event start time and the other one for the end time, this times determine the duration of the event.

  This widget includes the following views: month, week and day, agents can scroll thru the pages by using the right and left arrows.

  As mentioned before just enabling the widget is not enough, a couple of "Date/Time" dynamic fields for tickets should be added into the system (via Dynamic Fields link in "Admin" panel) and set them up in the SysConfig for this widget, both Dynamic Fields should be configured to be displayed on the ticket creation screens, they should be filled during ticket creation or any other ticket action screen (e.g. Free Fields) to describe the time frame for the calendar event (start and end time), the ticket zoom screen might be configured to show this dynamic fields also, in case you consider it necessary.

  Further configurations for this widget could be found under the "Frontend::Agent::Dashboard::EventsTicketCalendar" SubGroup in the SysConfig:

  - CalendarWidth

    Defines the calendar width in percent. Default is 95%.

  - DynamicFieldStartTime

    Defines dynamic field name for start time.

  - DynamicFieldEndTime

    Defines dynamic field name for end time.

  - Очереди

    Only the tickets on the queues specified in this setting will be considered in the calendar view.

  - DynamicFieldsForEvents

    Defines the dynamic fields that will be displayed in the calendar event overlay windows.

  - TicketFieldsForEvents

    Defines the ticket attributes that will be displayed in the calendar event overlay windows.

- Ticket Queue Overview

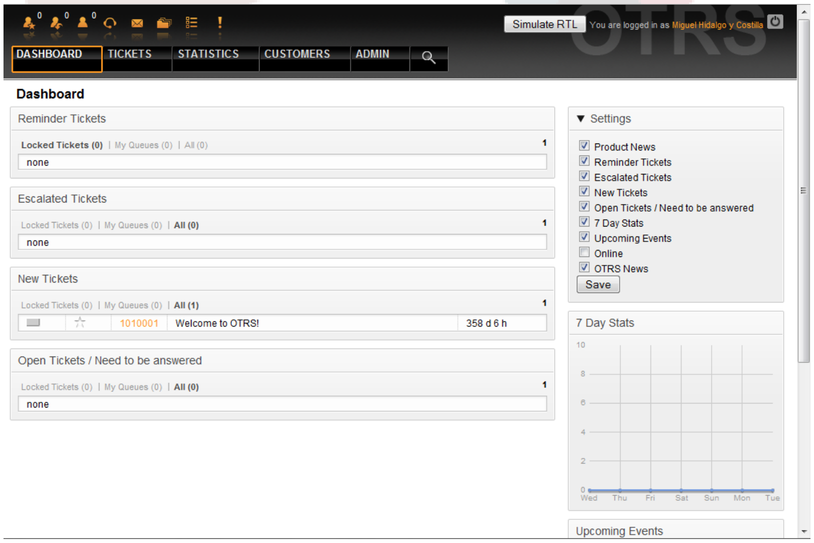
  This widget shows in a ticket count matrix where the rows represents queues and the columns represents the ticket states, then on each cell the number of tickets on a defined state that belongs on a particular queue is displayed.

  The widget also shows a Totals row and a Totals column, the Totals row shows the sum of the tickets for each state on all presented queues, while the Totals column represent the sum of the tickets for each queue on all presented states.

  The queues and states that are presented can be changed via Sysconfig.

  By clicking any of the ticket count numbers a ticket search results page will opened letting agents to have a more detailed overview of them.

In the right column is located an special widget that allow you to control the widgets you want to show or hide, This is the Settings widget. Click on it's header to expand the section and see all available widgets, as shown in Figure. Each widget name has a checkbox, Use this checkboxes to define the visibility of the widgets in the dashboard (unchecked widgets will not be shown) after you define the visibility options and click on 'Save' for storing your changes. This section is fixed on the screen, this means you can't drag and drop it, or close it.



You are logged in as Miguel Hidalgo y Costilla

Dashboard

Reminder Tickets

Locked Tickets (0) | My Queues (0) | All (0) 1

none

Escalated Tickets

Locked Tickets (0) | My Queues (0) | All (0) 1

none

New Tickets

Locked Tickets (0) | My Queues (0) | All (1) 1

| ☆ | 1010001 | Welcome to OTRS! | 358 d 6 h |

Open Tickets / Need to be answered

Locked Tickets (0) | My Queues (0) | All (0) 1

none

▼ Settings

☑ Product News
☑ Reminder Tickets
☑ Escalated Tickets
☑ New Tickets
☑ Open Tickets / Need to be answered
☑ 7 Day Stats
☑ Upcoming Events
☐ Online
☑ OTRS News

Save

7 Day Stats

Upcoming Events

Рисунок: Настройки Панели Управления.

Right column dashboard widgets.

- 7 Day Stats

  It shows a graph of ticket activity over the past 7 days that includes 2 lines. One that is usually blue color, represents the amount of created tickets per day and the second one, usually orange and represents the closed tickets per day.

- Upcoming Events

  Tickets on short for escalating or already escalated are listed here, info from this widget is very helpful since you have the chance to know about tickets needs your attention and you can decide in which ones you want to focus your effort on, set priorities or simply check what's coming on.

- OTRS News

  A complete list about OTRS activities and so important information about new product releases or patches.

- Online

  Here is showed a summary about the current agents logged the system, it also includes a section for customers on-line, please notice this widget is normally hidden, it can be shown using Settings widget described before.

7. Что такое очередь?

На физическом уровне для почтового ящика создается специальный файл (иногда его еще называют почтовый спул), в который помещаются и хранятся все сообщения. Новые сообщения электронной почты просто добавляются в конец этого файла. Почтовая программа, которая используется для чтения и написания писем (такие программы еще называются пользовательскими почтовыми агентами - MUA (Mail User Agent) читает этот файл и отображает его содержимое.

Очереди OTRS можно сравнить с Inbox-файлом, в котором хранятся входящие сообщения электронной почты, так как в них тоже может хранится множество сообщений. Очереди имеют много функций, схожих с теми, которые присущи Inbox-файлу. Как агенты так и пользователи, все должны помнить о том, в какой очереди содержится заявка. Агенты могут открывать, редактировать заявки, размещенные в очереди, а также, перемещать их из одной очереди в другую. Но зачем перемещать заявки?

Чтобы обьяснить это с более практической точки зрения давайте вспомним пример с компанией Максима, которая была описана в разделе пример системы обработки заявок. Максим установил OTRS чтобы избавится от существующего хаоса в системе поддержки и предоставить возможность своей команде лучше управлять службой поддержки клиентов, покупающих видеомагнитофоны.

One queue holding all requests is enough for this situation. However, after some time Max decides to also sell DVD recorders. Now, the customers have questions not only about the video recorder, but also about the new product. More and more emails get into the single queue of Max's OTRS and it's difficult to have a clear picture of what's happening.

Max decides to restructure his support system, and adds two new queues. So now three queues are being used. New messages arriving at the ticket system are stored into the old queue titled "raw". Of the two new queues, one titled "video recorder" is exclusively for video recorder requests, while the other one titled "dvd recorder" is exclusively for dvd recorder requests.

Max asks Sandra to watch the "raw" queue and sort (dispatch) the messages either into "video recorder" or "dvd recorder" queue, depending on the customer request. John only has access to the "video recorder" queue, while Joe can only answer tickets in the "dvd recorder" queue. Max is able to edit tickets in all queues.

OTRS supports access management for users, groups, and roles, and it is easy to setup queues that are accessible only to some user accounts. Max could also use another way to get his requests into the different queues, with filter rules. Otherwise, if two different mail addresses are used, Sandra only has to dispatch those emails into the two other queues, which can't be dispatched automatically.

Сортировка входящих сообщений в различные очереди позволяет сохранять систему поддержки в структурированном и аккуратном виде. Поскольку ваши агенты находятся в разных группах и имеют различные права для доступа к очередям, система может быть оптимизирована еще больше. Очереди могут быть использованы для определения структуры всей компании а также для определения рабочего потока процессов. Максим может к примеру создать еще одну очередь и назвать ее "продажи", которая будет содержать подочереди "запросы", "предложения", "заказы", "оплаты" и т.д. Такой подход поможет Максиму структурировать и оптимизировать процесс работы с транзакциями.

Чем лучше структурированна система поддержки, тем меньше времени требуется для различных задач. Это приводит к уменьшению рабочего времени, соответственно и к уменьшению денежных затрат. Очереди могут помочь оптимизировать бизнес-процессы вашей компании.

8. What is the queue overview?

The queue overview offers a view of all queues in which tickets are present, and for which the user has RW permissions.

The queue overview offers a variaty of options for daily work with OTRS. The first of theese is the My Queue. In the Agent Preferences, or when administering agents, a set of queuse can be defined for which the agent has been assigned to work within. All the tickets will appear in this default view, when accessing the Ticket -> Queue View menu.

The second option offered by the Queue View is a drill down navigation into individual queues and sub-queues containing tickets to be worked upon.

In both of the view types, the user also has the added ability to see either all unlocked tickets (this is the default filter), or the user can then chooose to view all available tickets. Tickets must be in one of the viewable state types to be shown in the queue view. Per default, these are 'open, new, pending reminder, pending auto'.

There are visual alarms, to aid the user.

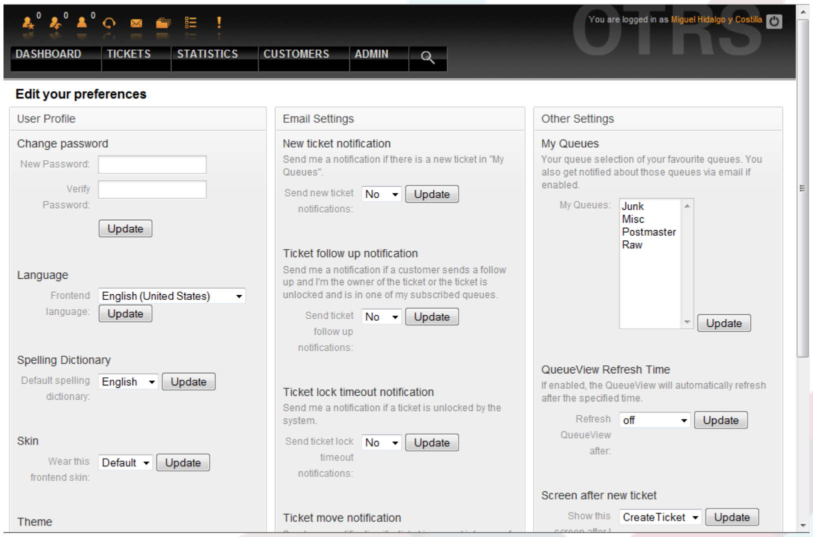
Visual Alarms

- Highlight Age 1: Sets the age in minutes (first level) for highlighting queues that contain untouched tickets. Seen in above in the "Raw" queue.
- Highlight Age 2: Sets the age in minutes (second level) for highlighting queues that contain untouched tickets. Seen in above in the "Postmaster" queue.

- Blink: Activates a blinking mechanism of the queue that contains the oldest ticket. Not supported in all browsers. In that case it will appear red, as seen in above in the "Junk" queue.
- Bold: The current queue will be bolded, as seen above in the "My Queues".

9. Пользовательские настройки

OTRS users such as customers, agents and the OTRS administrator can configure their account preferences as per their needs. Agent can access the configuration screen by clicking on their login name at the top right corner of the web interface (see Figure below), and customers must click on the "Preferences" link (see Figure below).



You are logged in as Miguel Hidalgo y Costilla

Edit your preferences

User Profile

Change password

New Password:

Verify Password:

Language

Frontend language: English (United States)

Spelling Dictionary

Default spelling dictionary: English

Skin

Wear this frontend skin: Default

Theme

Email Settings

New ticket notification

Send me a notification if there is a new ticket in "My Queues".

Send new ticket notifications: No

Ticket follow up notification

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.

Send ticket follow up notifications: No

Ticket lock timeout notification

Send me a notification if a ticket is unlocked by the system.

Send ticket lock timeout notifications: No

Ticket move notification

Other Settings

My Queues

Your queue selection of your favourite queues. You also get notified about those queues via email if enabled.

My Queues: Junk
Misc
Postmaster
Raw

QueueView Refresh Time

If enabled, the QueueView will automatically refresh after the specified time.

Refresh QueueView after: off

Screen after new ticket

Show this screen after I: CreateTicket

Figure: Agent's personal preferences.

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

Профиль пользователя

- Изменение текущего пароля.
- Выбор языка интерфейса.
- Переключение между темами интерфейса.
- Изменение темы веб-интерфейса.
- Активация и настройка функции "время вне оффиса".

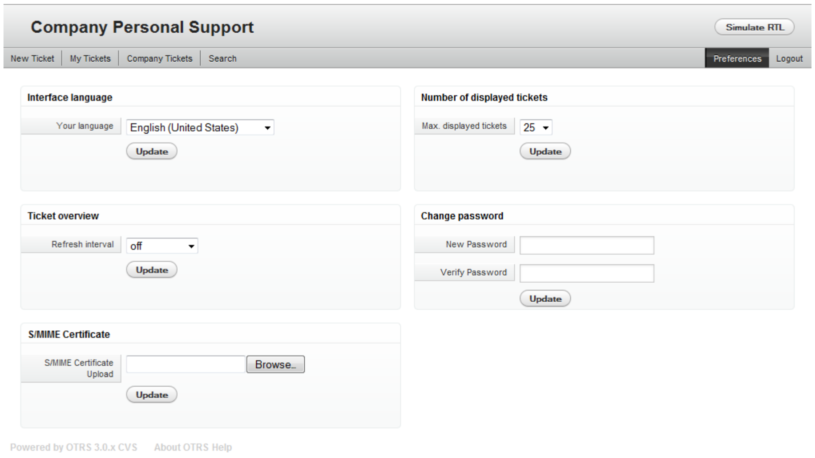
Настройки электронной почты

- Выберите события, при наступлении которых, система должна отсылать уведомления агентам на их емейлы.

Другие настройки

- В меню "Мои Очереди" выберите те очедеди, которые хотите отслеживать более внимательно.

- Установите время обновления страницы для шаблона отображения очередей.
- Выберите страницу, которая должна отображатся после создания заявки.



Interface language

Your language: English (United States)

Number of displayed tickets

Max. displayed tickets: 25

Ticket overview

Refresh interval: off

Change password

New Password:

Verify Password:

S/MIME Certificate

S/MIME Certificate Upload:

Figure: Customer's personal preferences.

Клиент может выбрать язык веб-интерфейса системы, изменять время обновления страницы в шаблоне просмотра очередей, выбирать максимальное количество отображаемых заявок. Кроме этого также можно установить новый пароль.

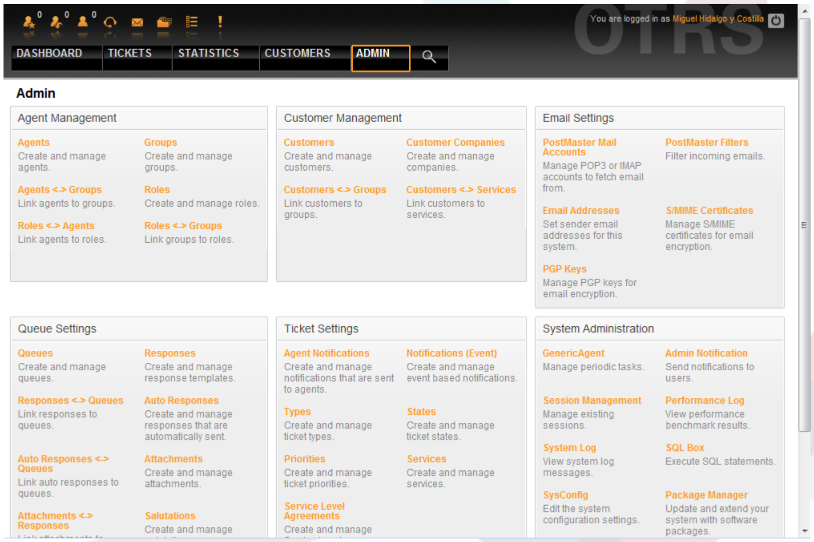
Глава 4. Администрирование

1. Панель управления OTRS

1.1. Основы

The following system configuration settings are available to OTRS administrators by accessing the Admin page of the OTRS web interface - adding agents, customers and queues, ticket and mail settings, installing additional packages such as FAQ and ITSM, and much more.

Agents who are members of the *admin* group can access the Admin area by clicking the *Admin* link in the navigation bar (see Figure below). Agents without sufficiently elevated access rights will not be able to access this link.



Admin

Agent Management

Agents
Create and manage agents.

Groups
Create and manage groups.

Agents <-> Groups
Link agents to groups.

Roles
Create and manage roles.

Roles <-> Agents
Link agents to roles.

Roles <-> Groups
Link groups to roles.

Customer Management

Customers
Create and manage customers.

Customer Companies
Create and manage companies.

Customers <-> Groups
Link customers to groups.

Customers <-> Services
Link customers to services.

Email Settings

PostMaster Mail Accounts
Manage POP3 or IMAP accounts to fetch email from.

PostMaster Filters
Filter incoming emails.

Email Addresses
Set sender email addresses for this system.

S/MIME Certificates
Manage S/MIME certificates for email encryption.

PGP Keys
Manage PGP keys for email encryption.

Queue Settings

Queues
Create and manage queues.

Responses
Create and manage response templates.

Responses <-> Queues
Link responses to queues.

Auto Responses
Create and manage responses that are automatically sent.

Auto Responses <-> Queues
Link auto responses to queues.

Attachments
Create and manage attachments.

Attachments <-> Responses

Salutations
Create and manage

Ticket Settings

Agent Notifications
Create and manage notifications that are sent to agents.

Notifications (Event)
Create and manage event based notifications.

Types
Create and manage ticket types.

States
Create and manage ticket states.

Priorities
Create and manage ticket priorities.

Services
Create and manage services.

Service Level Agreements
Create and manage

System Administration

GenericAgent
Manage periodic tasks.

Admin Notification
Send notifications to users.

Session Management
Manage existing sessions.

Performance Log
View performance benchmark results.

System Log
View system log messages.

SQL Box
Execute SQL statements.

SysConfig
Edit the system configuration settings.

Package Manager
Update and extend your system with software packages.

Рисунок: Страница OTRS-администратора.

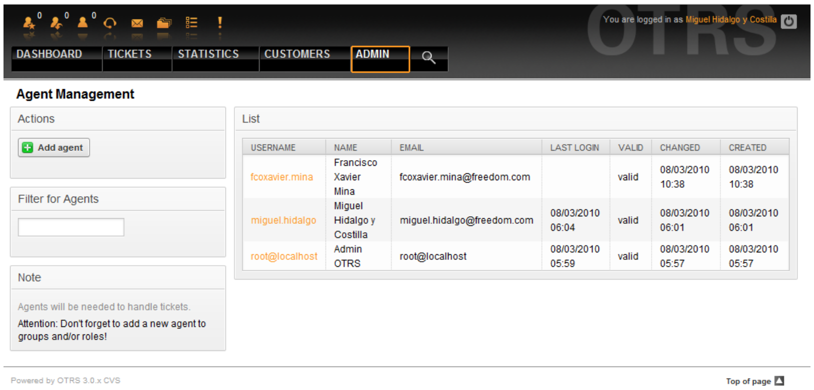
1.2. Агенты, Группы и Роли

1.2.1. Агенты

By clicking the link *Agents*, you get access to the agent management screen of OTRS (see Figure below). Administrators can add, change or deactivate agent accounts. Furthermore they can also manage agent preferences, including the language and notification settings for the individual agent's interface.

Примечание

Учетная запись (аккаунт) агента может быть деактивирована, но не удалена. Деактивация осуществляется путем установки параметра Действительный в значение *не действительный* или *не действительный-временно*.



Agent Management

List

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

Note

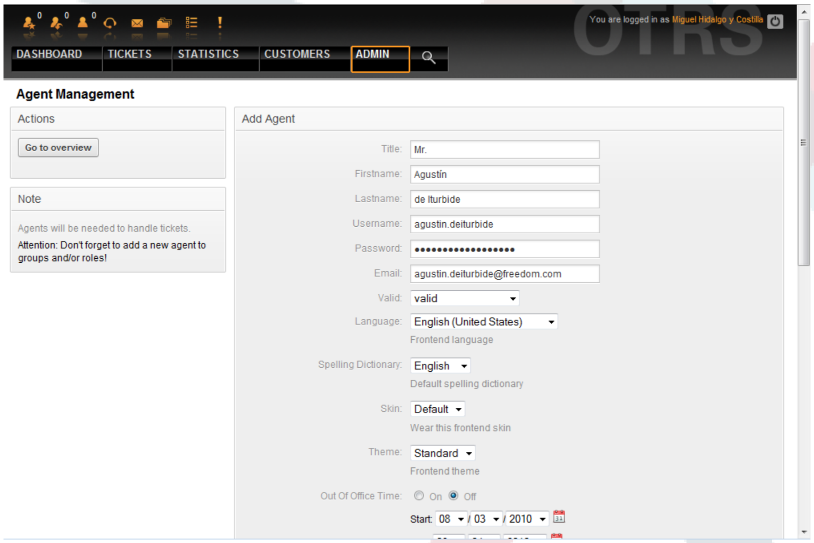
Agents will be needed to handle tickets.

Attention: Don't forget to add a new agent to groups and/or roles!

Powered by OTRS 3.0.x CVS

Рисунок: Управление Агентами.

To register an agent, click on the "Add agent" button, enter the required data and press the Submit button at the bottom of the screen, as shown in Figure.



Agent Management

Add Agent

Title:	Mr.
Firstname:	Agustín
Lastname:	de Iturbide
Username:	agustin.deiturbide
Password:	••••••••••••••••••
Email:	agustin.deiturbide@freedom.com
Valid:	valid
Language:	English (United States) Frontend language
Spelling Dictionary:	English Default spelling dictionary
Skin:	Default Wear this frontend skin
Theme:	Standard Frontend theme
Out Of Office Time:	○ On ● Off Start: 08 / 03 / 2010

Note

Agents will be needed to handle tickets.

Attention: Don't forget to add a new agent to groups and/or roles!

Рисунок: Добавление нового агента.

После того, как новая учетная запись агента создана, его нужно сделать членом одной или нескольких груп или ролей. За дополнительной информацией о группах и ролях обращайтесь к разделу Группы и Роли данной главы.

1.2.2. Группы

Каждая учетная запись агента должна принадлежать по крайней мере к одной группе или роли. После установки системы есть три предопределенных групы, которые приведены в Таблице 5-1.

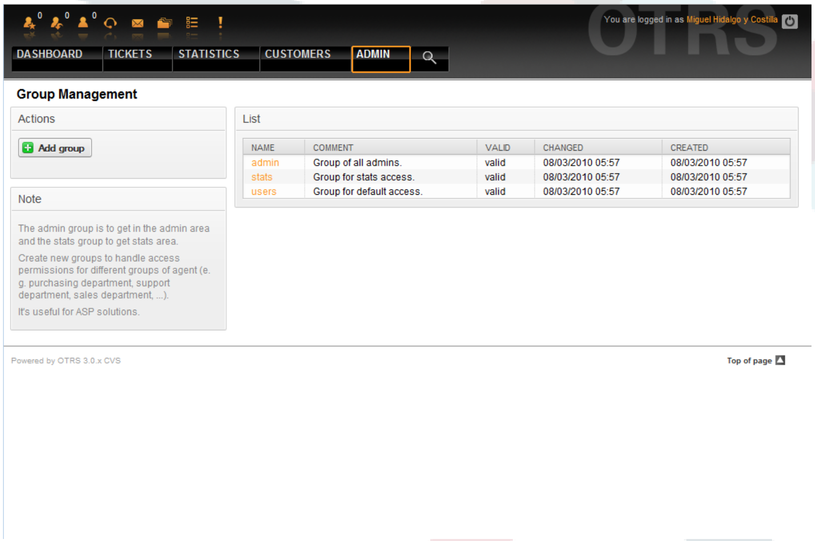
Таблица 4.1. Предопределенные групы, доступны для новой (только что) установленной системы OTRS

Группа	Описание
admin	Допускается для выполнения административных задач в системе.
stats	Компетентные для генерации статистики и доступа к модулю статистики OTRS.
users	Агенты должны входить в эту группу с правами для чтения и записи. Тогда они смогут получить доступ ко всем функциям системы обработки заявок.

Примечание

In a brand new OTRS installation, the *users* group initially does not have any members. The agent 'root@localhost' belongs by default to the admin and stats groups.

Чтобы перейти на страницу управления группами, нажмите в панели администрирования ссылку *Группы* (см. нижеприведенный Рисунок).



Group Management

Note

The admin group is to get in the admin area and the stats group to get stats area.

Create new groups to handle access permissions for different groups of agent (e. g. purchasing department, support department, sales department, ...).

It's useful for ASP solutions.

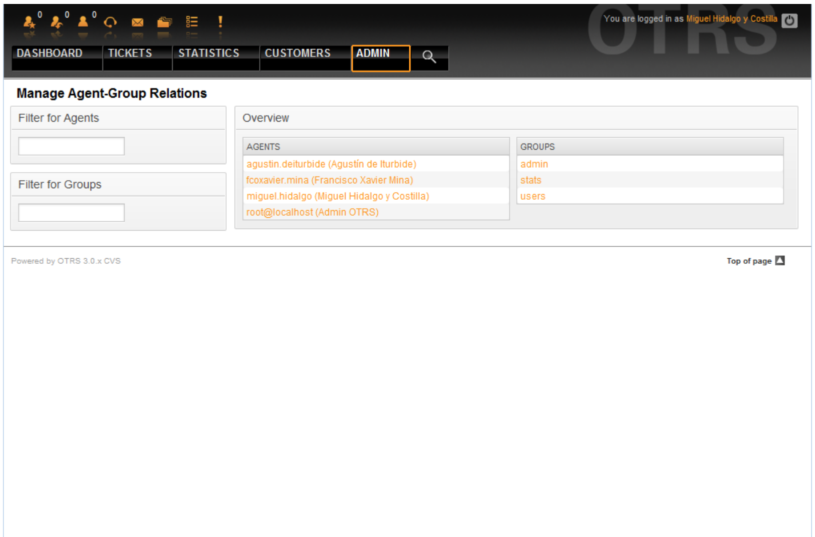
NAME	COMMENT	VALID	CHANGED	CREATED
admin	Group of all admins.	valid	08/03/2010 05:57	08/03/2010 05:57
stats	Group for stats access.	valid	08/03/2010 05:57	08/03/2010 05:57
users	Group for default access.	valid	08/03/2010 05:57	08/03/2010 05:57

Рисунок: Управление групами.

Примечание

Как в случае с агентами, группа в OTRS может быть только деактивирована, но не удалена. Деактивация осуществляется путем установки флага Действительный в значение *не действительный* или *не действительный-временно*.

Чтобы добавить агента в группу, или изменить приналежность агентов к группе, используйте ссылку на Панели Управления *Агенты <-> Группы* (см. нижеприведенный Рисунок).



You are logged in as Miguel Hidalgo y Costilla

Manage Agent-Group Relations

Filter for Agents

Filter for Groups

Overview

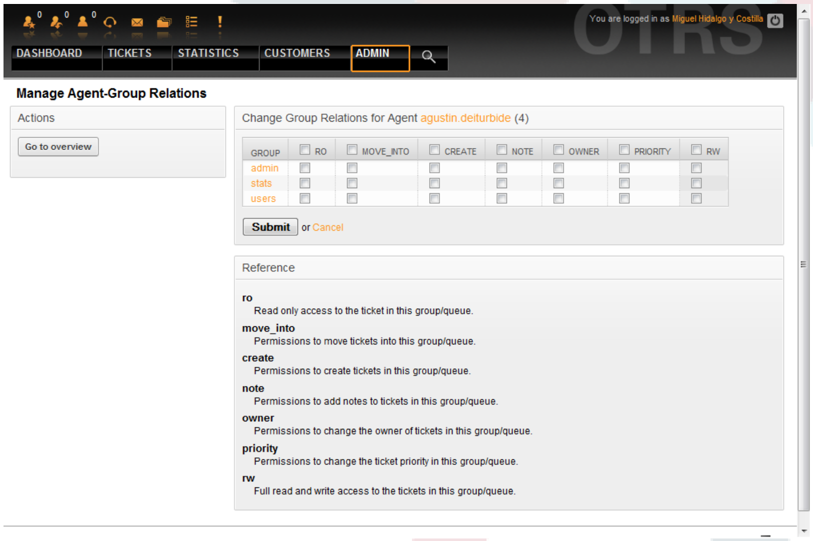
AGENTS	GROUPS
agustin.deiturbide (Agustín de Iturbide)	admin
fcoxavier.mina (Francisco Xavier Mina)	stats
miguel.hidalgo (Miguel Hidalgo y Costilla)	users
root@localhost (Admin OTRS)	

Powered by OTRS 3.0.x CVS

Top of page

Рисунок: Управление групами.

An overview of all groups and agents in the system is displayed on this page. You can also use the available filters to find a specific entity. If you want to change the groups that an agent is a member of, just click on the agent's name (see Figure below). To change the agents associated with a group, just click on the group you want to edit (see Figure below).



You are logged in as Miguel Hidalgo y Costilla

Manage Agent-Group Relations

Actions

Go to overview

Change Group Relations for Agent agustin.deiturbide (4)

GROUP	□ RO	□ MOVE_INTO	□ CREATE	□ NOTE	□ OWNER	□ PRIORITY	□ RW
admin	□	□	□	□	□	□	□
stats	□	□	□	□	□	□	□
users	□	□	□	□	□	□	□

Submit or Cancel

Reference

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

move_into
Permissions to move tickets into this group/queue.

create
Permissions to create tickets in this group/queue.

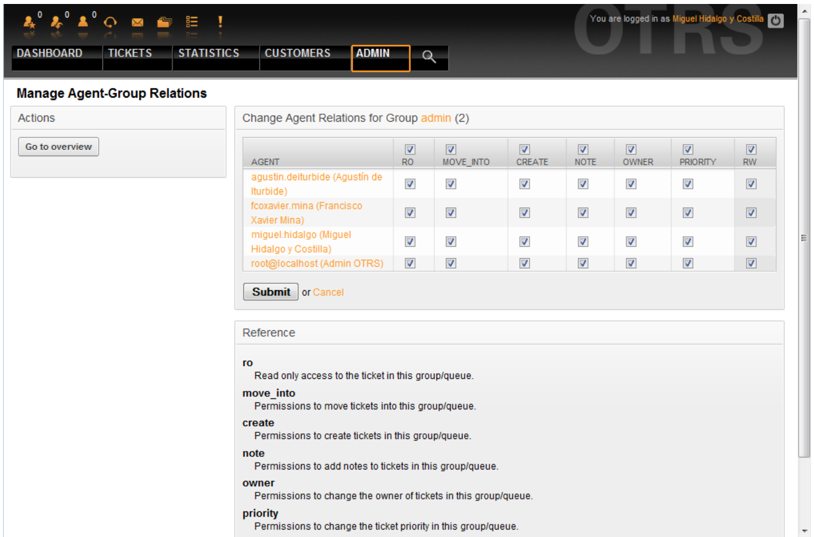
note
Permissions to add notes to tickets in this group/queue.

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

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

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

Рисунок: Изменение группы, к которой принадлежит агент.



Manage Agent-Group Relations

Actions

Change Agent Relations for Group admin (2)

AGENT	☑ RO	☑ MOVE_INTO	☑ CREATE	☑ NOTE	☑ OWNER	☑ PRIORITY	☑ RW
agustin.deiturbide (Agustín de Iturbide)	☑	☑	☑	☑	☑	☑	☑
fcoxavier.mina (Francisco Xavier Mina)	☑	☑	☑	☑	☑	☑	☑
miguel.hidalgo (Miguel Hidalgo y Costilla)	☑	☑	☑	☑	☑	☑	☑
root@localhost (Admin OTRS)	☑	☑	☑	☑	☑	☑	☑

Reference

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

move_into
Permissions to move tickets into this group/queue.

create
Permissions to create tickets in this group/queue.

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

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

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

Рисунок: Изменение агентов, входящих в определенную группу.

Each group has a set of rights associated with it, and each group member (agent) may have some combination of these rights for themselves. A list of the permissions / rights is shown in Table 5-2.

Таблица 4.2. Права, связанные с OTRS-группами

Права	Описание
ro	Права "только для чтения" заявок, записей и очередей этой группы.
move into	Права для перемещения заявок или записей между очередями или для области, принадлежащей группе.
create	Права для создания заявок или записей в очередях или области группы.
owner	Права для изменения владельца заявок или записей в очередях принадлежащих этой группе.
priority	Права для изменения приоритета заявок или записей в очереди или области, принадлежащей этой группе.
rw	Полный доступ к заявкам или записям в очереди или области, принадлежащей к этой группе.

Примечание

По умолчанию, СписокОчередей (QueueView) отображает только заявки в очередях, к которым агент имеет права *rw*, т.е к заявкам, с которыми он должен работать. Если вы хотите изменить это поведение, вы можете установить параметр Ticket::Frontend::AgentTicketQueue###ViewAllPossibleTickets в значение *Да*.

Not all available permissions are shown by default. These additional permissions can be added.

Таблица 4.3. Additional permission groups

Права	Описание
stats	Gives access to the stats page.
bounce	The right to bounce an email message (with bounce button in ticketZoom).
compose	The right to compose an answer for a ticket.
customer	The right to change the customer of a ticket.
forward	The right to forward a messages (with the forward button).
pending	The right to set a ticket to pending.
phone	The right to add a phonecall to a ticket.
responsible	The right to change the responsible agent for a ticket.

Примечание

These permissions can be added by changing the Ticket::Core::Ticket::System::Permission

1.2.3. Роли

Roles are a powerful feature to manage the access rights of many agents in a very simple and quick manner. They are particularly useful for large, complex support systems with a lot of agents, groups and queues. An example below explains when they should be used.

Suppose that you have a system with 100 agents, 90 of them with access to a single queue called "support" where all support requests are handled. The "support" queue contains multiple sub queues. The other 10 agents have permission to access all queues of the system. These 10 agents dispatch tickets, watch the raw queue and move spam messages into the "junk" queue.

The company now opens a new department that sells some products. Order request and acceptance, order confirmation, bills, etc. must be processed, and some of the company's agents are supposed to do this using OTRS. The different agents have to get access to the new queues that must be created.

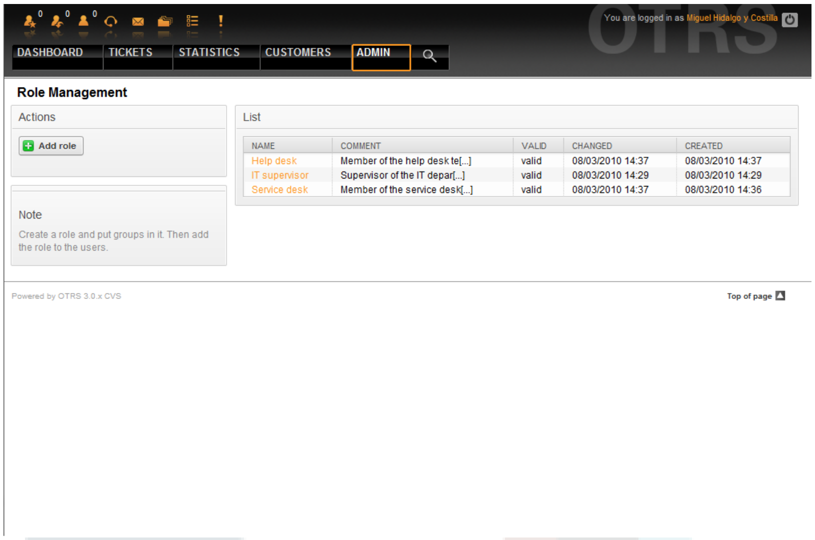
Because it would take a long time to change the access rights for the individual agents manually, roles that define the different access levels can be created. The agents can then be added to one or more roles, with their access rights being modified automatically. If a new agent account is created, it is also possible to add this account to one or more roles.

Примечание

Roles are really useful when dealing with complex organizations and when maintaining larger OTRS installations. Proper care is advised though. Mixing Agent to Group with Agent to Role mappings can make for a complex access control scheme, that is difficult to understand and maintain. If you wish to use only roles and disable the Agents <-> Groups option in the Admin area, you can do so by modifying the

Frontend::Module###AdminUserGroup in the SysConfig. Be aware that this won't remove already existing Agents to Group assignments!

Для управления ролями перейдите по ссылке *Роли* в Панели Администрирования (см. нижеприведенный Рисунок).



You are logged in as Miguel Hidalgo y Costilla

Role Management

Actions

List

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

Note

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

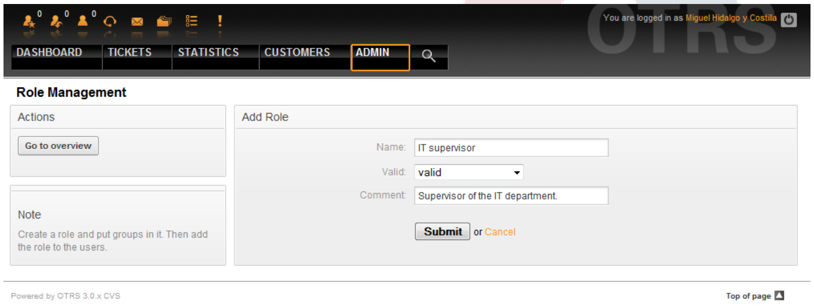
Powered by OTRS 3.0.x CVS

Рисуно: Управление Ролями.

Примечание

Как в случае с агентом и группами, однажды созданные роли могут быть деактивированы, но не удалены. Чтобы отключить, установите опцию Действительный в значение *не действительный* или *не действительный-временно*.

Приведен обзор всех ролей в системе. Нажмите на имя роли, чтобы отредактировать ее настройки. По умолчанию, в новой, только что установленной системе OTRS нету предопределенных ролей. Чтобы создать новую, нажмите кнопку "Добавить роль", введите все необходимые данные и нажмите кнопку Отправить (см. нижеприведенный Рисунок).



You are logged in as Miguel Hidalgo y Costilla

Role Management

Actions

Add Role

Name: IT supervisor

Valid: valid

Comment: Supervisor of the IT department.

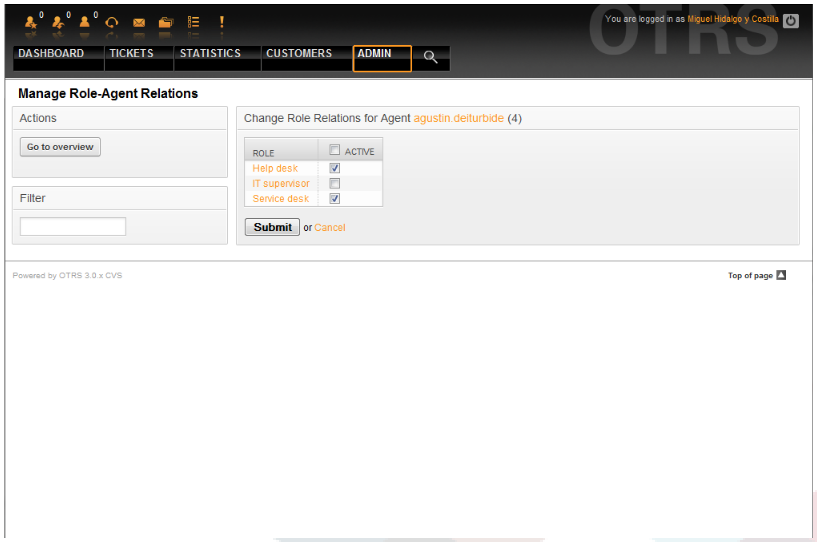
Note

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

Powered by OTRS 3.0.x CVS

Рисунок: Добавление новой роли.

Чтобы получить список всех ролей и агентов в системе, перейдите по ссылке Роли <-> Агенты в Панели Администрирования. Также, для поиска конкретного элемента можно использовать фильтры. Если необходимо изменить роли связанные с определенным агентом, просто кликните на имя агента (см. нижеприведенный Рисунок). Чтобы изменить агентов, связанных с ролью, нажмите на роль, которую нужно отредактировать (см. нижеприведенный Рисунок).



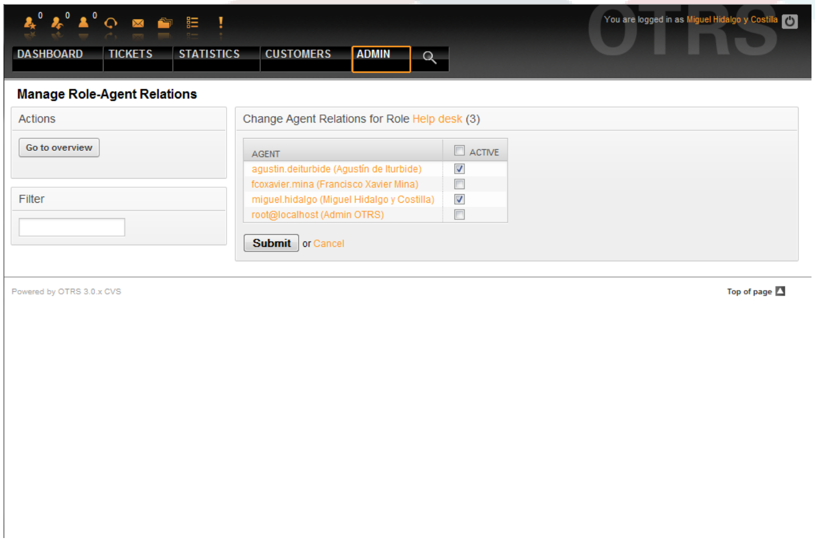
Manage Role-Agent Relations

Change Role Relations for Agent agustin.deiturbide (4)

ROLE	□ ACTIVE
Help desk	☑
IT supervisor	□
Service desk	☑

Powered by OTRS 3.0.x CVS

Figure: Изменение Ролей, связанных с Агентом.



Manage Role-Agent Relations

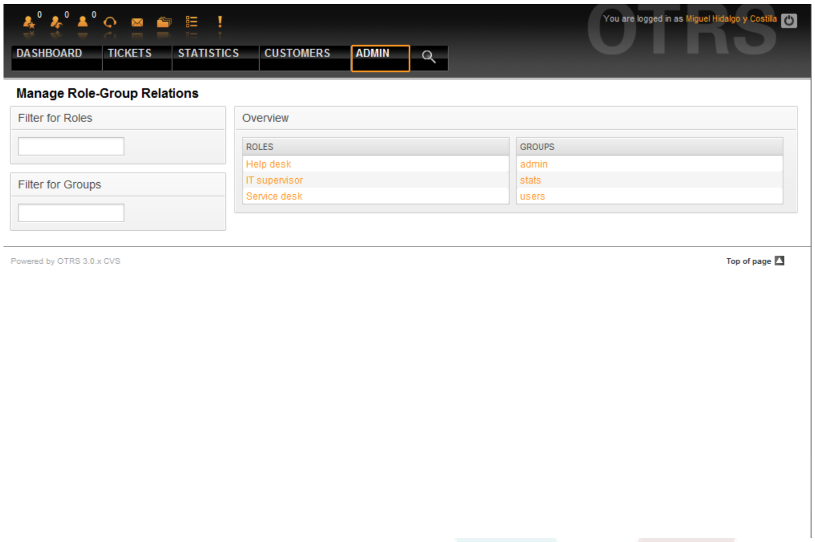
Change Agent Relations for Role Help desk (3)

AGENT	□ ACTIVE
agustin.deiturbide (Agustín de Iturbide)	☑
fcoxavier.mina (Francisco Xavier Mina)	□
miguel.hidalgo (Miguel Hidalgo y Costilla)	☑
root@localhost (Admin OTRS)	□

Powered by OTRS 3.0.x CVS

Figure: Изменение Агентов, связанных с конкретной Ролью.

Чтобы получить список всех ролей и групп системы, перейдите по ссылке Роли <-> Группы на Панели Управления. На экране появится страница, приведенная ниже на Рисунке. Для поиска конкретного элемента также можно использовать фильтры.



Manage Role-Group Relations

Filter for Roles

Filter for Groups

Overview

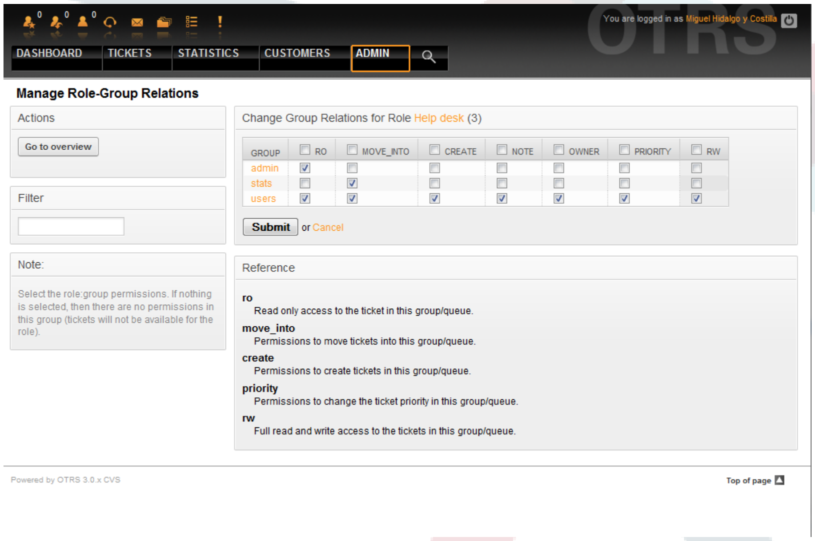
ROLES	GROUPS
Help desk	admin
IT supervisor	stats
Service desk	users

Powered by OTRS 3.0.x CVS

Top of page

Figure: Управление отношениями Роли-Группы.

Чтобы определить различные права доступа для роли, нажмите на имя роли или группы (см. ниже Рисунок 5.13 и 5.14, соответственно).



Manage Role-Group Relations

Actions

Go to overview

Filter

Note:

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

Change Group Relations for Role Help desk (3)

GROUP	□ RO	□ MOVE_INTO	□ CREATE	□ NOTE	□ OWNER	□ PRIORITY	□ RW
admin	☑	□	□	□	□	□	□
stats	□	☑	□	□	□	□	□
users	☑	☑	☑	☑	☑	☑	☑

Submit or Cancel

Reference

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

move_into
Permissions to move tickets into this group/queue.

create
Permissions to create tickets in this group/queue.

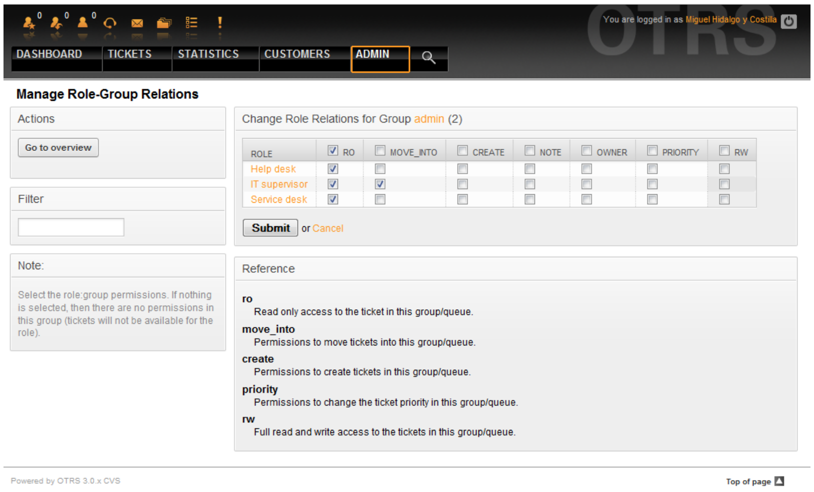
priority
Permissions to change the ticket priority in this group/queue.

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

Powered by OTRS 3.0.x CVS

Top of page

Рисунок: Изменение связти между Группой и Ролью.



You are logged in as Miguel Hidalgo y Costilla

Manage Role-Group Relations

Actions

Go to overview

Filter

Note:

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

Change Role Relations for Group admin (2)

ROLE	☑ RO	□ MOVE_INTO	□ CREATE	□ NOTE	□ OWNER	□ PRIORITY	□ RW
Help desk	☑	□	□	□	□	□	□
IT supervisor	☑	☑	□	□	□	□	□
Service desk	☑	□	□	□	□	□	□

Submit or Cancel

Reference

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

move_into
Permissions to move tickets into this group/queue.

create
Permissions to create tickets in this group/queue.

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

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

Powered by OTRS 3.0.x CVS

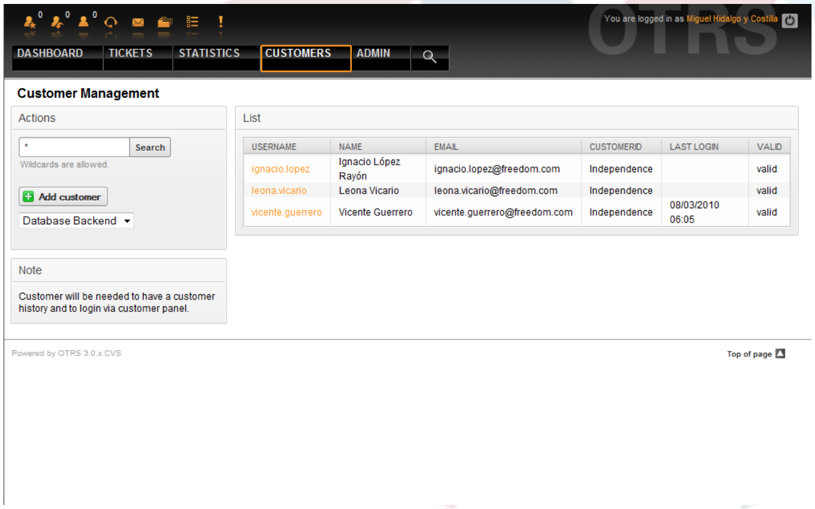
Top of page

Рисунок: Изменение отношения Роли к Группе.

1.3. Клиенты и Группы Клиентов

1.3.1. Клиенты

OTRS supports different types of users. Using the link "Customers" (via the navigation bar, or the Admin page), you can manage the accounts of your customers (see Figure below), who can log into the system via the Customers interface (customer.pl). Through this interface, your customers can not only create tickets but also review their past tickets for new updates. It is important to know that a customer is needed for the ticket history in the system.



You are logged in as Miguel Hidalgo y Costilla

Customer Management

Actions

* Search

Wildcards are allowed.

Add customer

Database Backend

Note

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

List

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

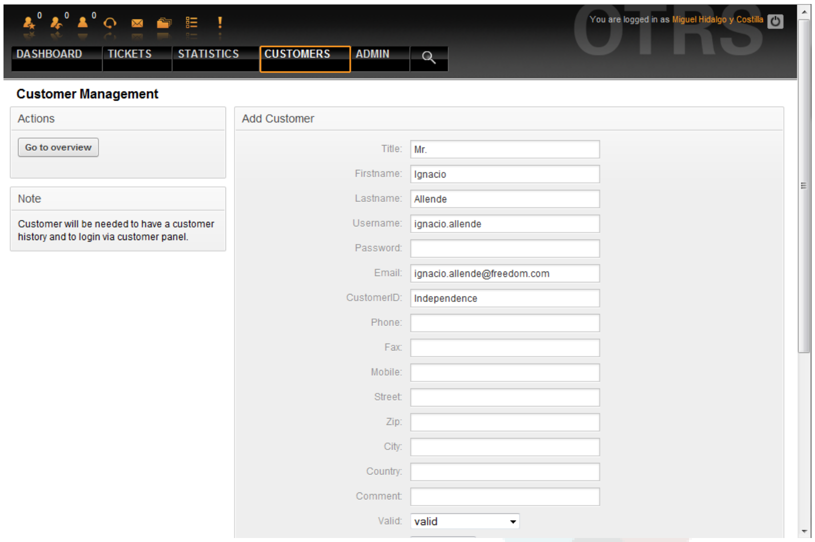
Powered by OTRS 3.0.x CVS

Top of page

Рисунок: Управление клиентами.

Нажав на имя клиента, по зарегистрированным клиентам можно производить поиск, редактировать их данные. Также есть возможность изменить клиентский back-end, за более подробной информацией обращайтесь к главе внешние хранилища данных.

Чтобы создать новую учетную запись клиента, нажмите кнопку "Добавить клиента" (см. нижеприведенный Рисунок). Некоторые поля являются обязательными, т.е они должи содержать данные, поэтому если вы оставите их пустыми, они будут выделены красным цветом.



Customer Management

Note

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

Add Customer

Title:	Mr.
Firstname:	Ignacio
Lastname:	Allende
Username:	ignacio.allende
Password:	
Email:	ignacio.allende@freedom.com
CustomerID:	Independence
Phone:	
Fax:	
Mobile:	
Street:	
Zip:	
City:	
Country:	
Comment:	
Valid:	valid

Рисунок: Добавление клиента.

Клиент может получить доступ к системе путем ввода логина и пароля. CustomerID нужен для идентификации клиента и его заявок. Поскольку адресс электронной почты является уникальным, его можно использовать в качестве уникального идетнтификатора (ID).

Примечание

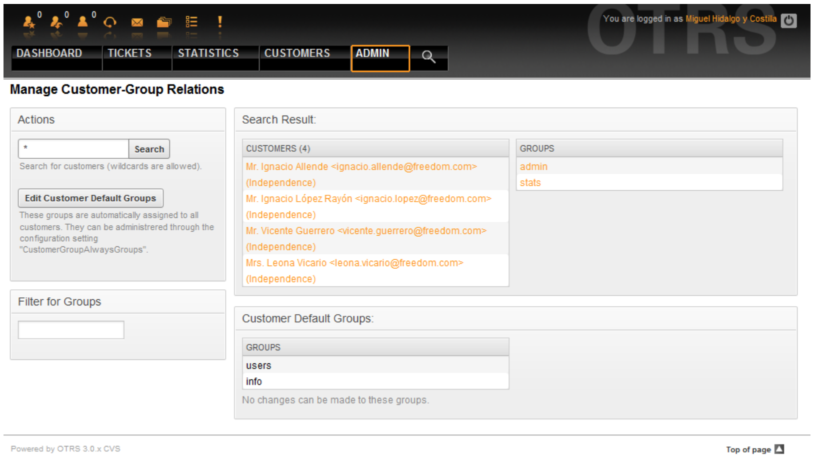
Как и в случае с агентами, группами и ролями, система не позволяет удалять клиентов, а только деактивировать их, установив опцию Действительный в значение *недействительный* или *недействительный-временно*.

1.3.2. Группы Клиентов

Клиенты также могут быть добавлены в группу, что может быть полезно, если вы хотите обьеденить нескольких клиентов той же компании для доступа к одной или нескольким очередям. Сначала создайте группу, в которую будут входить клиенты, для этого перейдите по ссылке Модуль Управления Группами . Затем добавьте очереди и выберите новую группу для этих очередей.

Следующим шагом будет включение поддержки клиентов группы. Это можно сделать используя конфигурационный параметр CustomerGroupSupport в SysConfig. Используя параметр CustomerGroupAlwaysGroups можно указать группы по умолчанию, к которым будет принадлежать новый пользователь, таким образом каждая новая учетная запись будет добавлятся в эту группу.

Перейдя по ссылке "Клиенты <-> Группы" вы можете управлять отношениями конкретного пользователя к разным группам (см. нижеприведенный Рисунок).



You are logged in as Miguel Hidalgo y Costilla

Manage Customer-Group Relations

Actions

Search for customers (wildcards are allowed).

These groups are automatically assigned to all customers. They can be administrered through the configuration setting "CustomerGroupAlwaysGroups".

Filter for Groups

Search Result:

CUSTOMERS (4)	GROUPS
Mr. Ignacio Allende <ignacio.allende@freedom.com> (Independence)	admin
Mr. Ignacio López Rayón <ignacio.lopez@freedom.com> (Independence)	stats
Mr. Vicente Guerrero <vicente.guerrero@freedom.com> (Independence)	
Mrs. Leona Vicario <leona.vicario@freedom.com> (Independence)	

Customer Default Groups:

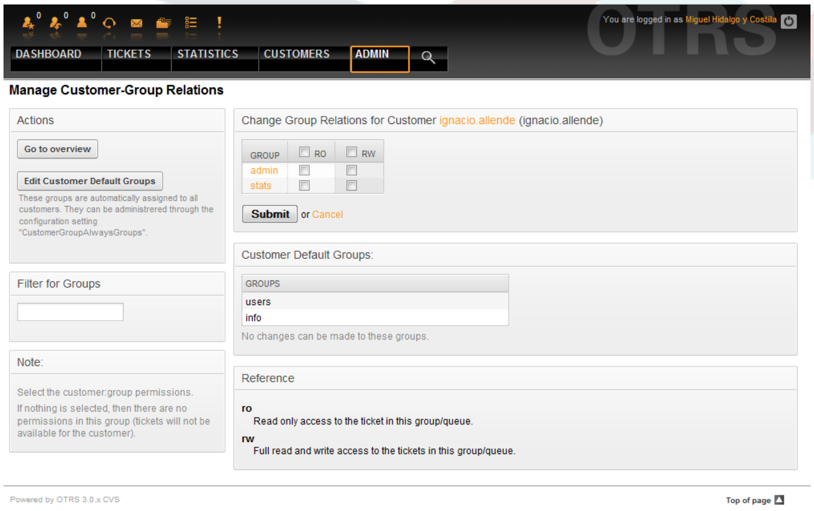
GROUPS
users
info

No changes can be made to these groups.

Powered by OTRS 3.0.x CVS

Рисунок: Управление отношениями Клиент-Группа.

Для определения различных групп, в которые должен входить клиент, и наоборот, нажмите на соответствующее имя пользователя или группы (см. ниже Рисунок 5.16 и 5.17, соответственно).



You are logged in as Miguel Hidalgo y Costilla

Manage Customer-Group Relations

Actions

These groups are automatically assigned to all customers. They can be administrered through the configuration setting "CustomerGroupAlwaysGroups".

Filter for Groups

Note:

Select the customer:group permissions.

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

Change Group Relations for Customer ignacio.allende (ignacio.allende)

GROUP	□ RO	□ RW
admin	□	□
stats	□	□

Customer Default Groups:

GROUPS
users
info

No changes can be made to these groups.

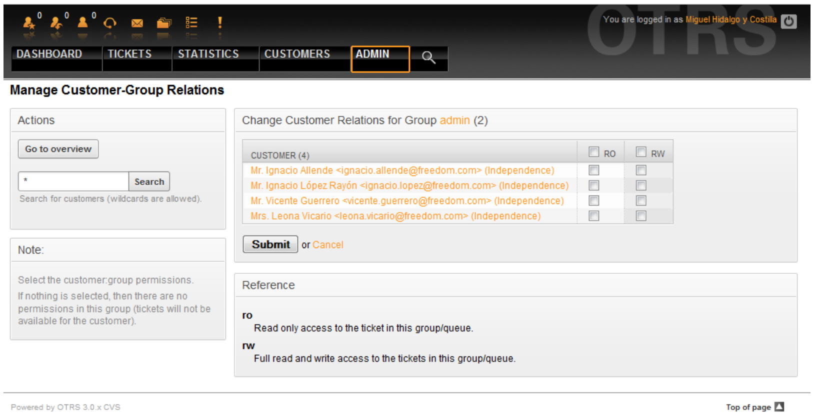
Reference

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

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

Powered by OTRS 3.0.x CVS

Рисунок: Изменение связи между Группой и Клиентом.



You are logged in as Miguel Hidalgo y Costilla

Manage Customer-Group Relations

Actions

*

Search for customers (wildcards are allowed).

Note:

Select the customer:group permissions.

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

Change Customer Relations for Group admin (2)

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

Submit or Cancel

Reference

ro

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

rw

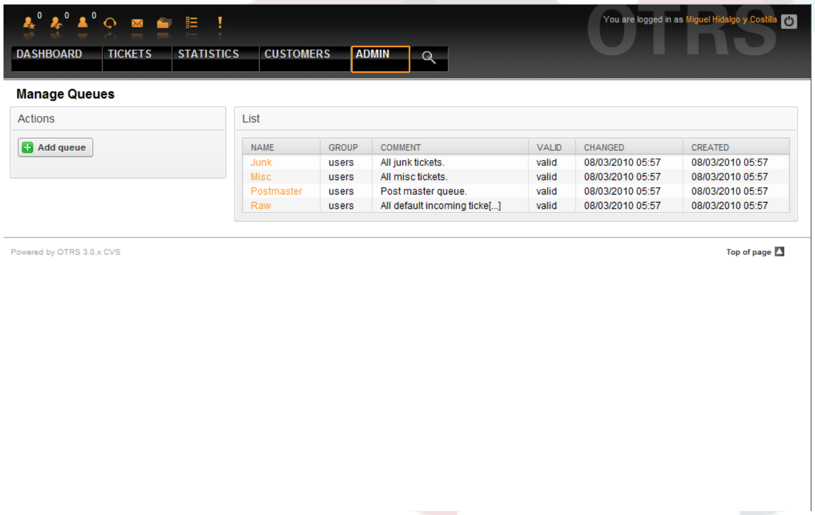
Full read and write access to the tickets in this group/queue.

Powered by OTRS 3.0.x CVS

Рисунок: Изменение связи между Клиентом и Группой.

1.4. Очереди

Используя ссылку "Очереди", находящуюся в Панели Администрирования, можете управлять очередями системы (см. нижеприведенный Рисунок). В новой, только установленной системе OTRS существует 4 предопределенных очереди: Junk, Misc, Postmaster и Raw. Если не используются ранее определенные фильтры и правила, все входящие сообщения сохраняются в очереди "Raw". Очередь "Junk" используется для хранения спам-сообщений.



You are logged in as Miguel Hidalgo y Costilla

Manage Queues

Actions

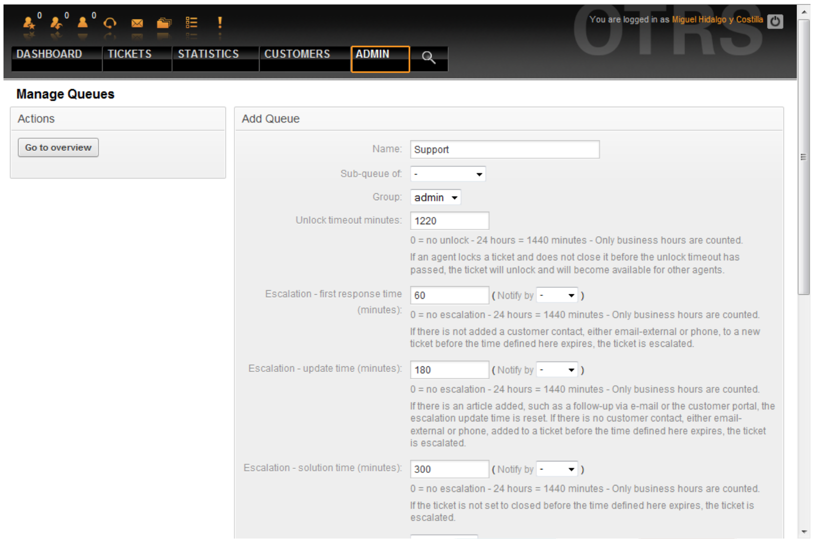
List

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

Powered by OTRS 3.0.x CVS

Рисунок: Управление очередями.

Здесь можно добавлять и корректировать очереди (см. нижеприведенный Рисунок). Можно определить группу, которая должна использовать очередь. Также очередь можно установить как под-очередь уже существующей очереди.



Manage Queues

Add Queue

Name: Support

Sub-queue of: -

Group: admin

Unlock timeout minutes: 1220

0 = no unlock - 24 hours = 1440 minutes - Only business hours are counted.

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.

Escalation - first response time (minutes): 60 (Notify by -)

0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.

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

Escalation - update time (minutes): 180 (Notify by -)

0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.

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 (minutes): 300 (Notify by -)

0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.

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

Рисунок: Добавление новой очереди.

Для очереди можно определить время разблокировки - если агент заблокировал заявку и не закрыл ее до истечения времени блокировки, заявка будет автоматически разблокирована и доступна для других агентов.

Есть три типа настроек эскалации времени, которые могут быть связаны на уровне очередей:

Эскалация - Время Первого Отклика

- After creation of the ticket, if the time defined here expires without any communication with the customer, either by email or phone, the ticket is escalated.

Эскалация - Время обновления

- If there is a customer followup either via e-mail or the customer portal, that is recorded in the ticket, the escalation update time is reset. If there is no customer contact before the time defined here expires, the ticket is escalated.

Эскалация - Время решения

- Если до истечения оговореного времени заявка не закрыта - она эскалируется.

Используя опцию 'Ticket lock after a follow-up', можно определить будет ли заявка установлена в значение 'заблокирована' для старого владельца, если она была закрыта а потом открыта заново. Такое поведение гарантирует, что последующая заявка будет обрабатываться агентом, который работал с этой заявкой ранее.

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

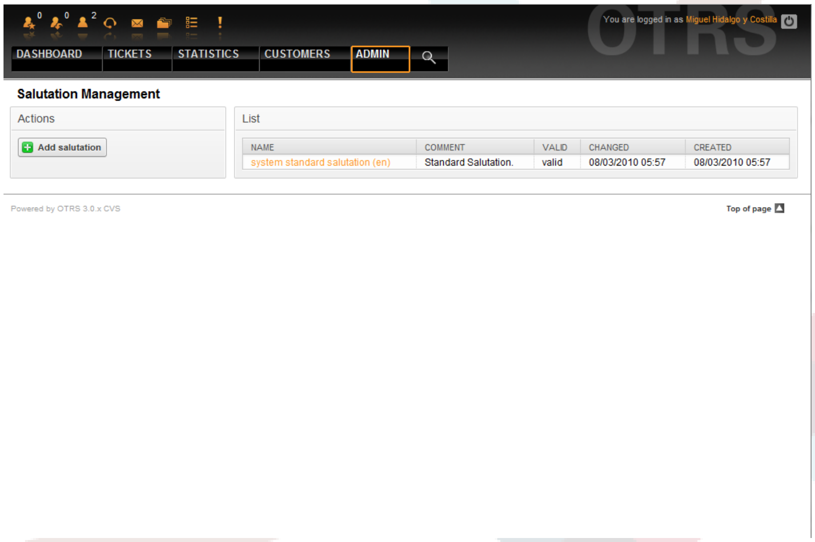
Примечание

Как в случае с агентами, группами и клиентами, очереди не могут быть удалены, а только отключены, путем установки опции "Действительная" в значение *недействительная* или *недействительная-временно*.

1.5. Salutations, signatures, attachments and templates

1.5.1. Приветсвтия

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



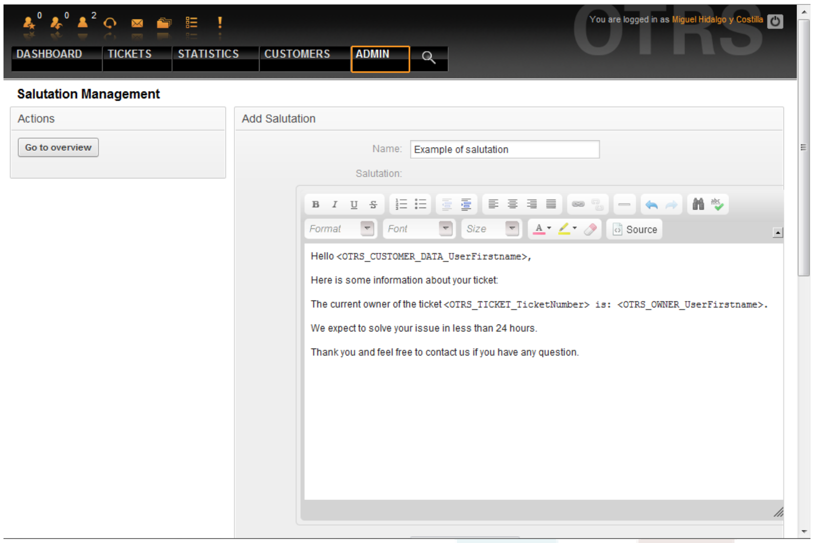
Salutation Management

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

Рисунок: Управление приветсвиями.

После установки в системе есть одно предопределенное приветствие: "system standard salutation (en)".

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



Salutation Management

Add Salutation

Name: Example of salutation

Salutation:

Hello <OTRS_CUSTOMER_DATA_UserFirstname>,

Here is some information about your ticket:

The current owner of the ticket <OTRS_TICKET_TicketNumber> is: <OTRS_OWNER_UserFirstname>.

We expect to solve your issue in less than 24 hours.

Thank you and feel free to contact us if you have any question.

Рисунок: Добавление нового приветствия.

В приветствиях можно использовать переменные. Когда вы отвечаете на заявку, имена переменных будут заменены на их значения.

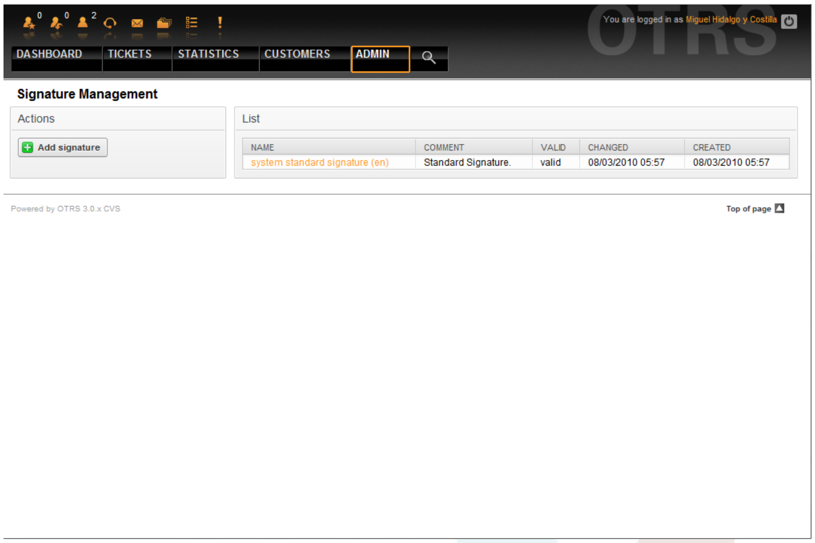
The different variables you can use in templates are listed in the lower part of the salutation screen. If you use, for example, the variable <OTRS_LAST_NAME> the last name of the ticket's sender will be included in your reply.

Примечание

Как и в случае с другими сущностями системы OTRS, приветствия не можно удалять, только деактивировать их, установив параметр Действительный в значение *не действительный* или *не действительный-временно*.

1.5.2. Подписи

Another text module for a template is the signature. Signatures can be linked to a queue, as described in the section about the queues. Please note that a signature will only be appended to a template text, if it has previously been linked to a queue. You can manage the signatures in your system by accessing the "Signatures" link of the Admin page, (see Figure below).

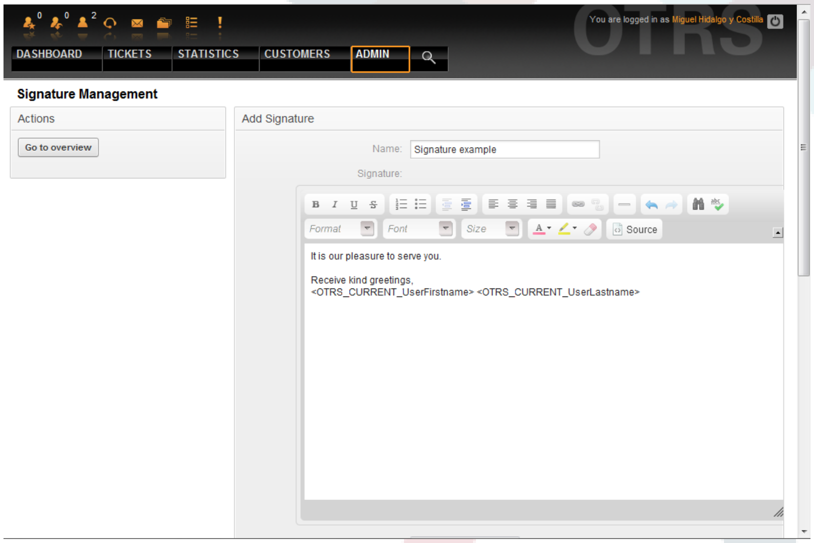


NAME	COMMENT	VALID	CHANGED	CREATED
system standard signature (en)	Standard Signature.	valid	08/03/2010 05:57	08/03/2010 05:57

Рисунок: Управление подписями.

В новой, только установленной системе OTRS есть одна предустановленная подпись: "system standard signature (en)".

Чтобы создать новую подпись, нажмите кнопку "Добавить подпись", введите все необходимы данные и нажмите кнопку "Отправить" (см. нижеприведенный Рисунок).



Name: Signature example

Signature:

It is our pleasure to serve you.

Receive kind greetings,
<OTRS_CURRENT_UserFirstname> <OTRS_CURRENT_UserLastname>

Рисунок: Добавление новой подписи.

Like salutations, signatures can also contain dynamic content, such as the first and last name of the agent who answers the ticket. Here too, variables can be used to replace the content of the signature text for every ticket. See the lower part of the signatures screen for the variables which can be used. If you include the variable <OTRS_LAST_NAME> in a signature for example, the last name of the agent who answers the ticket will replace the variable.

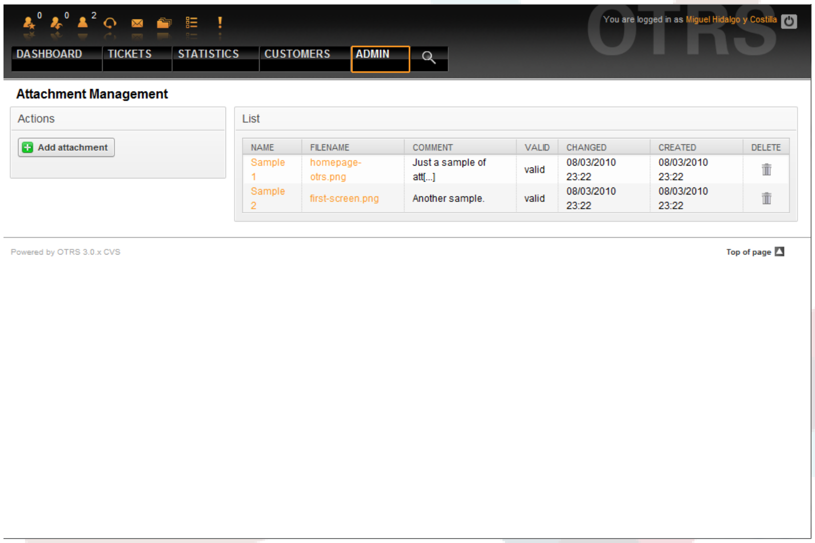
Примечание

Как и в случае с приветствиями, подписи не могут быть удалены, только деактивированы путем установки параметра Действительный в значения *не действительный* или *не действительный-временно*.

1.5.3. Вложения

You can also optionally add one or more attachments to a template. If the template is selected, the attachments will be attached to the message composition window. If necessary, the agent can remove the attachment from an individual template before sending it to the customer.

Используя ссылку "Вложения" на Панели Администрирования, можно загрузить вложения в базу данных системы (см. нижеприведенный Рисунок).



You are logged in as Miguel Hidalgo y Costilla

Attachment Management

Actions

Add attachment

List

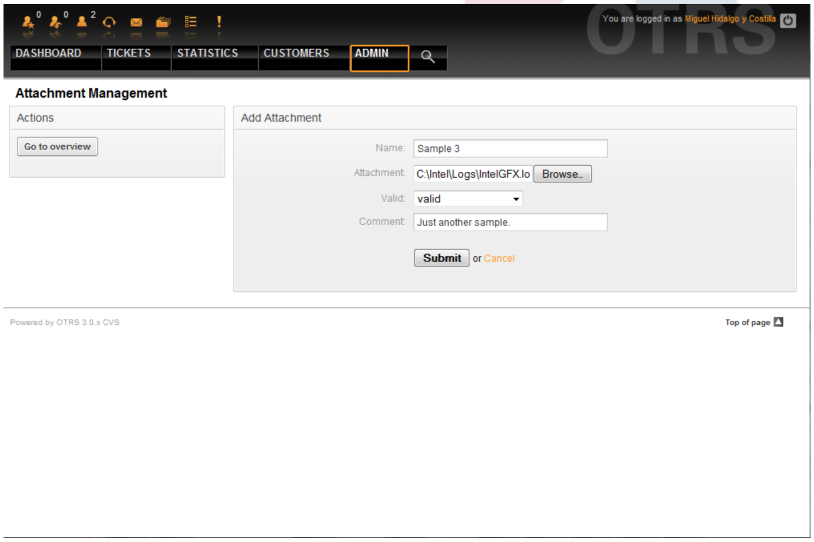
NAME	FILENAME	COMMENT	VALID	CHANGED	CREATED	DELETE
Sample 1	homepage-otrs.png	Just a sample of att[...]	valid	08/03/2010 23:22	08/03/2010 23:22	🗑
Sample 2	first-screen.png	Another sample.	valid	08/03/2010 23:22	08/03/2010 23:22	🗑

Powered by OTRS 3.0.x CVS

Top of page

Рисунок: Управление вложениями.

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



You are logged in as Miguel Hidalgo y Costilla

Attachment Management

Actions

Go to overview

Add Attachment

Name: Sample 3

Attachment: C:\Intel\Logs\IntelGFX.lo Browse...

Valid: valid

Comment: Just another sample.

Submit or Cancel

Powered by OTRS 3.0.x CVS

Top of page

Рисунок: Добавление нового вложения.

If an attachment is stored it can be linked to one or more templates. Click on the "Attachment <-> Templates" link of the Admin page (see Figure below).

Figure: Linking Attachments to Templates.

To associate different attachments with a specific template and vice versa, click on the corresponding template name or attachment (see below the Figures 5.27 and 5.28, respectively).

Figure: Change Attachment relations for a Template.

Figure: Change Template relations for an Attachment.

1.5.4. Templates

To speed up ticket processing and to standardize the look of answers, you can define templates in OTRS. A template can be linked to one or more queues and vice versa.

There are different kind of templates that are used in different parts of OTRS and they have its own purpose, the following is the list of possible template types:

- Answer: To be used as a ticket response or reply
- Create: To be used in New Phone or Email ticket
- Forward: To be used to forward an article to someone else
- PhoneCall: To be used in the Phone Call Inbound and Outbound screens

Answer templates can be accessed in two ways, from the ticket zoom screen in the article menu, or on a quicker way: from any ticket overview large screen such as Status View or Ticket View. For a fresh OTRS installation, the "empty answer" template (Answer) is set as the default for every queue.

As soon as Forward templates are added and assigned to queues, the "Forward" button in ticket zoom (that normally leads to a empty text forward screen) will change into a selection control, the selection is filled with the added Forward templates, by choosing one of the templates, the forward screen will be shown prefilled with the template text and attachments (similar to the reply selection box with the Answer templates).

Creating templates of type Create and PhoneCall will make visible the "Text Template" selection box in their respective screens, choosing a template for the list will populate the "Text" and "Attachment" fields (if available in the template). Notice that any previous change in the Text or attachments will be overwritten by selecting a template.

Clicking the "Templates" link on the Admin page brings you to the Template management screen (see Figure below).

Figure: Template management.

To create a new template, click on the "Add template" button, provide the required data (make sure to select the appropriate template type) and submit it (see Figure below).

Figure: Adding a template.

To add/remove templates to one or more queues, click on the "Templates <-> Queues" link on the Admin page (see Figure below). You can also use filters to get information regarding a specific entity.

Figure: Template-Queue relations management.

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

Figure: Change Queue relations for a Template.

Figure: Change Template relations for a Queue.

When choosing a template, additional information could be added to the template text, this depends on the template type:

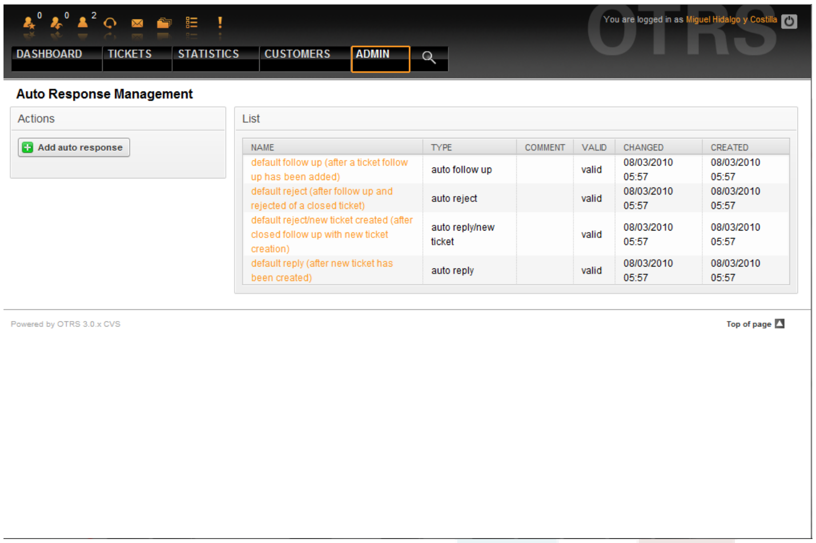
PhoneCall and Create templates does not add any content to the template text, however New Email Ticket screen adds the queue assigned signature to the resulting email body (this screen has a separated box to visualize the signature).

Answer templates text when selected also included the salutation associated with the queue, then the text of the template, then the quoted ticket text, and finally the signature associated with the queue.

Forward templates are similar to Answer templates, but they does not include the salutation part.

1.6. Авто ответы

OTRS allows you to send automatic responses to customers based on the occurrence of certain events, such as the creation of a ticket in a specific queue, the receipt of a follow-up message in regards to a ticket, the closure or rejection of a ticket, etc. To manage such responses, click the link "Auto responses" on the Admin page (see Figure below).

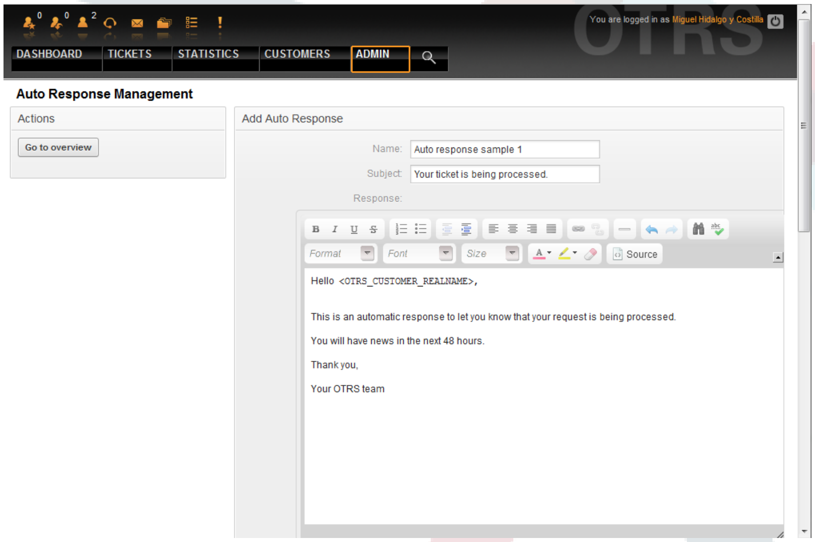


Auto Response Management

NAME	TYPE	COMMENT	VALID	CHANGED	CREATED
default follow up (after a ticket follow up has been added)	auto follow up		valid	08/03/2010 05:57	08/03/2010 05:57
default reject (after follow up and rejected of a closed ticket)	auto reject		valid	08/03/2010 05:57	08/03/2010 05:57
default reject/new ticket created (after closed follow up with new ticket creation)	auto reply/new ticket		valid	08/03/2010 05:57	08/03/2010 05:57
default reply (after new ticket has been created)	auto reply		valid	08/03/2010 05:57	08/03/2010 05:57

Рисунок: Управление Автоматическими ответами (авто-ответами).

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



Auto Response Management

Name: Auto response sample 1

Subject: Your ticket is being processed.

Response:

Hello <OTRS_CUSTOMER_REALNAME>,

This is an automatic response to let you know that your request is being processed.

You will have news in the next 48 hours.

Thank you,

Your OTRS team

Рисунок: Добавление Автоматического ответа (автоответа).

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

Для каждого авто-ответа можно указать специальное событе, при наступлении которого этот авто-ответ будет отсылатся. Предопределенные события, доступные после новой установки приведены в Таблицу 5-3.

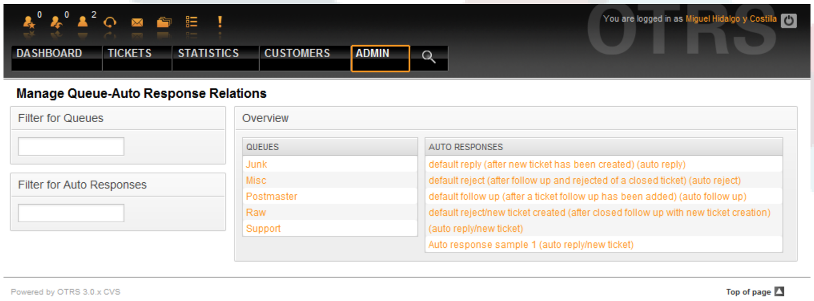
Таблица 4.4. События для Автоматических ответов

Название	Описание
auto reply	Создание заявки в определенной очереди.
auto reply/new ticket	Открытие уже закрытой заявки, например если клиент отвечает на определенную заявку.
auto follow up	Прием входящих заявок.
auto reject	Автоматический отказ от заявки, сделанный системой.
auto remove	Удаление заявко, завершенных системой.

Примечание

As with other OTRS entities, auto responses too cannot be deleted, only deactivated, by setting the Valid option to *invalid* or *invalid-temporarily*.

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

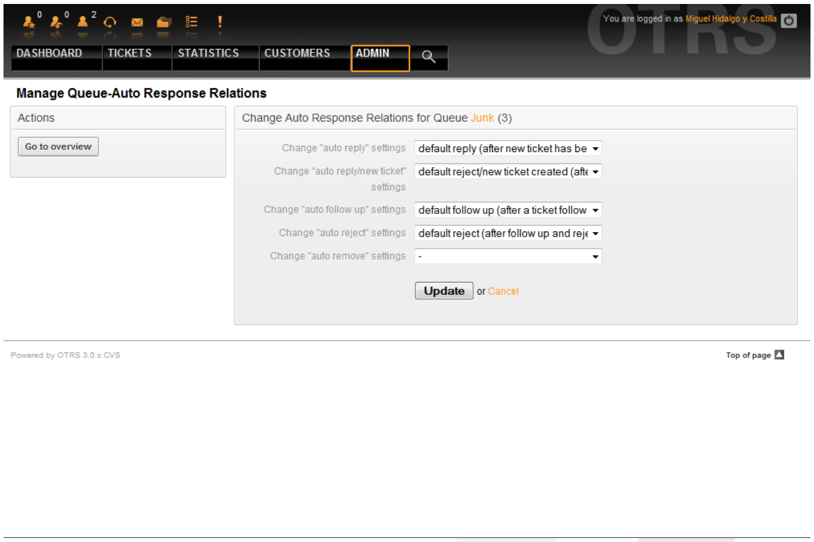

0
0
2
You are logged in as Miguel Hidalgo y Costilla
DASHBOARD
TICKETS
STATISTICS
CUSTOMERS
ADMIN
Filter for Queues
Filter for Auto Responses
Overview
Powered by OTRS 3.0.x CVS
Top of page

Manage Queue-Auto Response Relations

QUEUES	AUTO RESPONSES
Junk	default reply (after new ticket has been created) (auto reply)
Misc	default reject (after follow up and rejected of a closed ticket) (auto reject)
Postmaster	default follow up (after a ticket follow up has been added) (auto follow up)
Raw	default reject/new ticket created (after closed follow up with new ticket creation) (auto reply/new ticket)
Support	Auto response sample 1 (auto reply/new ticket)

Рисунок: Управление связью Очередь-Автоматический ответ.

To define the different auto responses that will be available for a queue, click on the corresponding queue name (see Figure below). It is also possible to edit an existing auto response - to do so, click on the response and edit in the same manner as editing a new auto response.



You are logged in as Miguel Hidalgo y Costilla

Manage Queue-Auto Response Relations

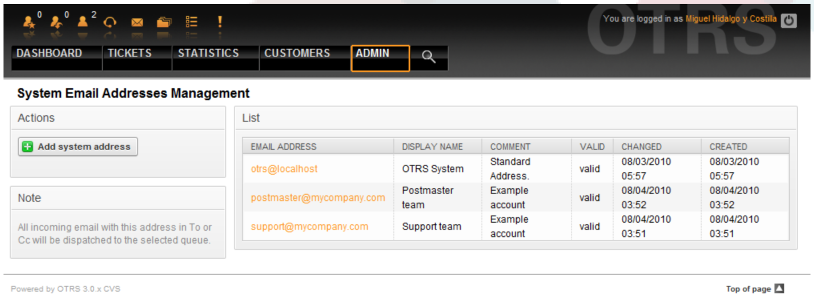
Change Auto Response Relations for Queue Junk (3)

Change "auto reply" settings: default reply (after new ticket has be
Change "auto reply/new ticket" settings: default reject/new ticket created (afte
Change "auto follow up" settings: default follow up (after a ticket follow
Change "auto reject" settings: default reject (after follow up and reje
Change "auto remove" settings: -

Рисунок: Изменение связи между Автоматическим Ответом и Очередью.

1.7. Адреса электронной почты

To enable OTRS to send emails, you need a valid email address to be used by the system. OTRS is capable of working with multiple email addresses, since many support installations need to use more than one. A queue can be linked to many email addresses, and vice versa. The address used for outgoing messages from a queue can be set when the queue is created. Use the "Email Addresses" link from the Admin page to manage all email addresses of the system (see Figure below).



You are logged in as Miguel Hidalgo y Costilla

System Email Addresses Management

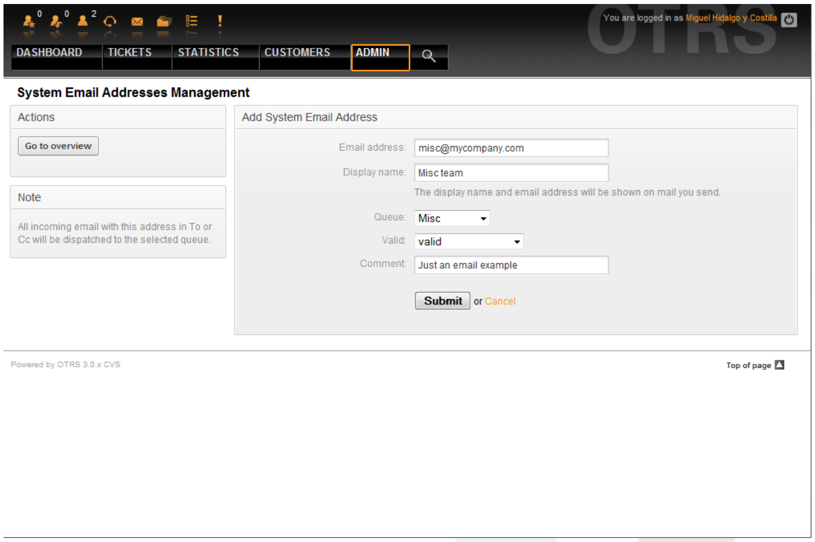
Note

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

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

Рисунок: Управление Системными Адресами Электронной Почты.

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



System Email Addresses Management

Actions

Note

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

Add System Email Address

Email address:	misc@mycompany.com
Display name:	Misc team
	The display name and email address will be shown on mail you send.
Queue:	Misc
Valid:	valid
Comment:	Just an email example

Рисунок: Добавление адреса электронной почты системы.

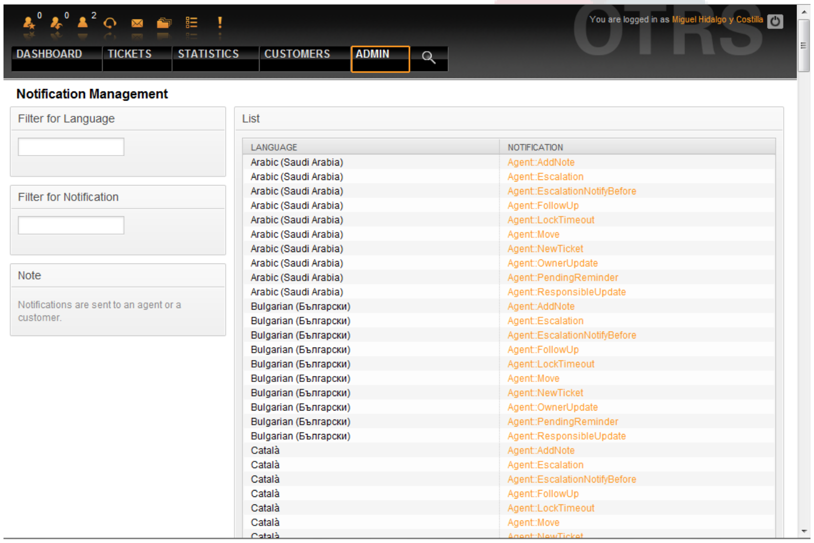
Примечание

Как и другие сущности OTRS, емейл адреса не могут быть удалены, только деактивированы, установив опцию Действительный в значение *не действительный* или *не действительный-временно*.

1.8. Уведомления

OTRS allows notifications to be sent to agents and customers, based on the occurrence of certain events. Agents can set the system events for their own notifications via the preferences link.

Through the "Agent Notifications" link on the Admin page, you can manage the notifications of your system (see Figure below). You can use filters to list only certain notifications.



Notification Management

Filter for Language

Filter for Notification

Note

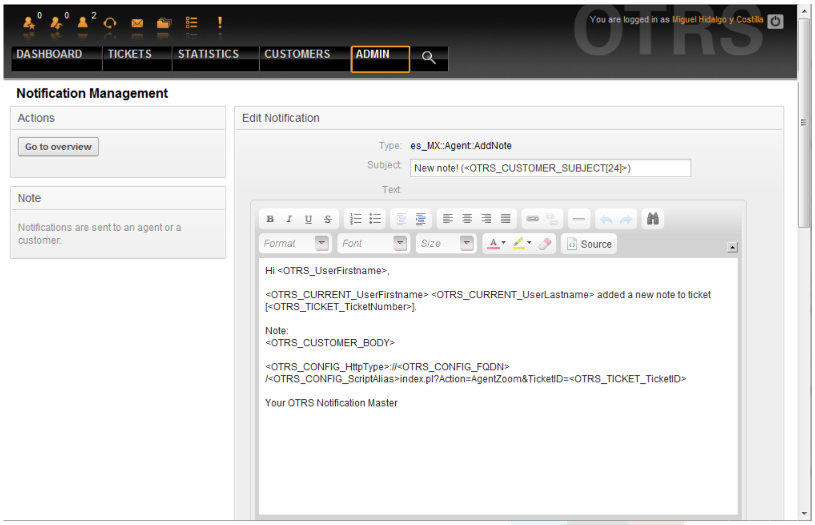
Notifications are sent to an agent or a customer.

List

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

Рисунок: Управление уведомлениями.

You can customize the subject and the text of the notifications. Click on the notification you want to change from the list, and its content will get loaded for editing (see Figure). Please note that there is a notification with the same name for each of the available languages.



Notification Management

Notifications are sent to an agent or a customer.

Edit Notification

Type: es_MX::Agent::AddNote

Subject: New note! (<OTRS_CUSTOMER_SUBJECT[24]>)

Text:

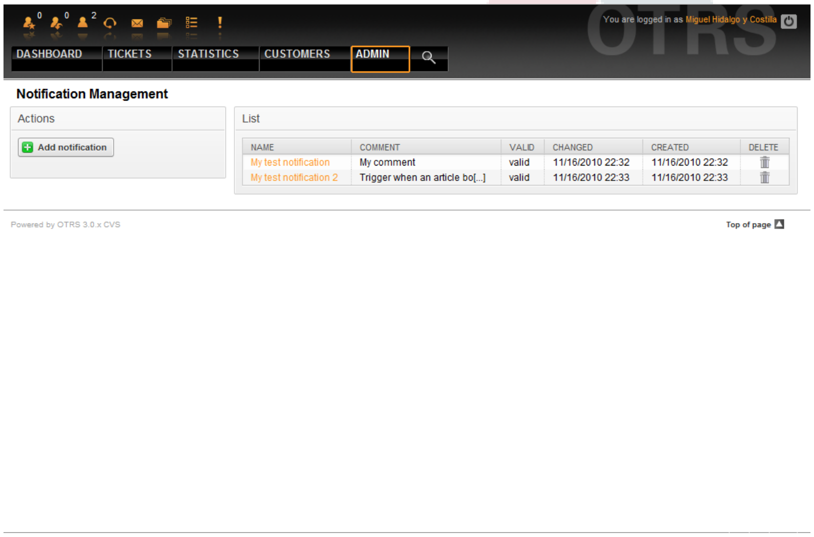
```
Hi <OTRS_UserFirstname>,

<OTRS_CURRENT_UserFirstname> <OTRS_CURRENT_UserLastname> added a new note to ticket
[<OTRS_TICKET_TicketNumber>].

Note:
<OTRS_CUSTOMER_BODY>

<OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>
/<OTRS_CONFIG_ScriptAlias>index.pl?Action=AgentZoom&TicketID=<OTRS_TICKET_TicketID>

Your OTRS Notification Master
```

Рисунок: Настройка уведомлений.

Just as with signatures and salutations, it is possible to dynamically create the content of a notification, by using special variables. You can find a list of variables at the bottom of the screen shown in the Figure.

Также есть возможность создавать уведомления основаные на системных событиях. Можно подробно указать когда и кому такое уведомление будет отправлено. Выбрать можно из большого количества параметров, таких как: группа(ы) получателя, агент(ы), роль(и), адресс(а) электронной почты, тип события, который генерирует уведомление, тип заявки, состояние, приоритет, очередь, блокировка, сервис, SLA и т.д.

In order to see a list of all event based notifications, click on the link "Notifications (Event)" on the Admin page (see Figure).



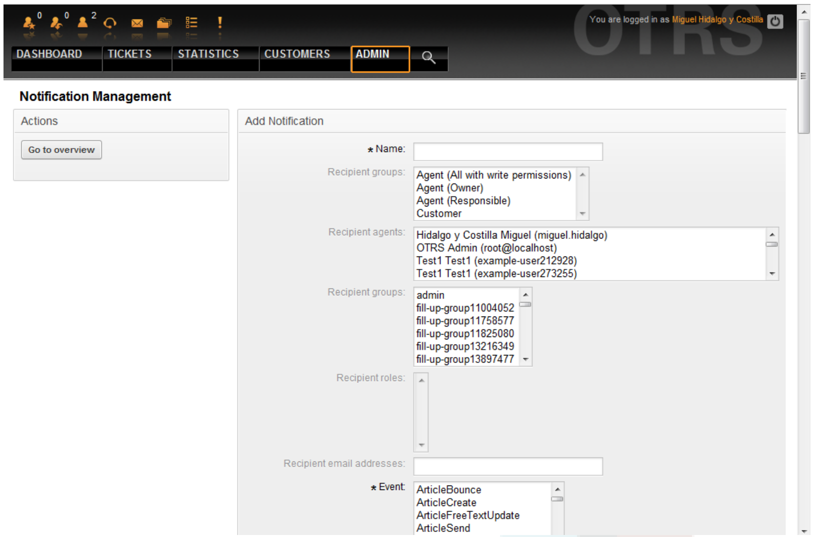
Notification Management

List

NAME	COMMENT	VALID	CHANGED	CREATED	DELETE
My test notification	My comment	valid	11/16/2010 22:32	11/16/2010 22:32	🗑
My test notification 2	Trigger when an article bo[...]	valid	11/16/2010 22:33	11/16/2010 22:33	🗑

Рисунок: Управления Уведомлениями, базирующихся на Событиях.

As shown in Figure, you can create a new notification by clicking on the Add button (see Figure).



You are logged in as Miguel Hidalgo y Costilla

Notification Management

Actions

Go to overview

Add Notification

* Name:

Recipient groups: Agent (All with write permissions), Agent (Owner), Agent (Responsible), Customer

Recipient agents: Hidalgo y Costilla Miguel (miguel.hidalgo), OTRS Admin (root@localhost), Test1 Test1 (example-user212928), Test1 Test1 (example-user273255)

Recipient groups: admin, fill-up-group11004052, fill-up-group11758577, fill-up-group11825080, fill-up-group13216349, fill-up-group13897477

Recipient roles:

Recipient email addresses:

* Event: ArticleBounce, ArticleCreate, ArticleFreeTextUpdate, ArticleSend

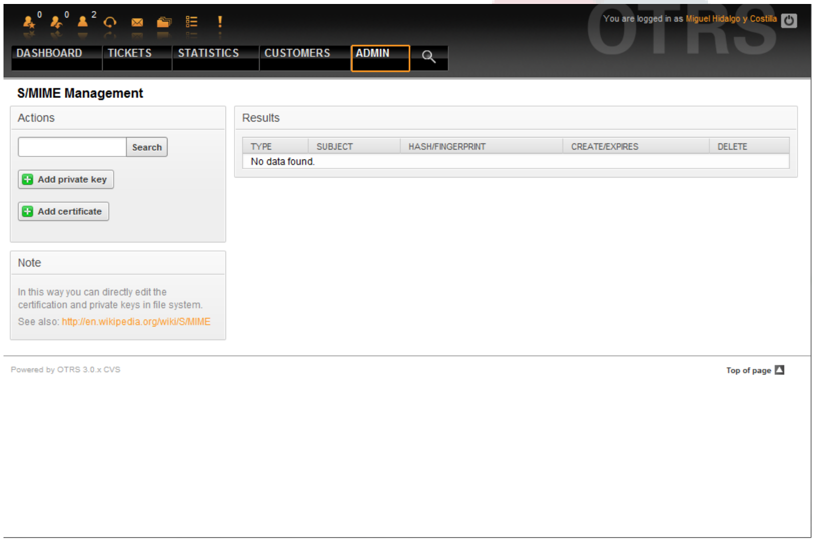
Рисунок: Регистрация Уведомления, основаного на Событии.

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.

1.9. SMIME

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

The "S/MIME Certificates" link on the Admin page allows you to manage your S/MIME certificates (see Figure below). You can add or remove certificates, and also search through the SMIME data.



You are logged in as Miguel Hidalgo y Costilla

S/MIME Management

Actions

Search

Add private key

Add certificate

Note

In this way you can directly edit the certification and private keys in file system.

See also: http://en.wikipedia.org/wiki/S/MIME

Results

TYPE	SUBJECT	HASH/FINGERPRINT	CREATE/EXPIRES	DELETE
No data found.				

Powered by OTRS 3.0.x CVS

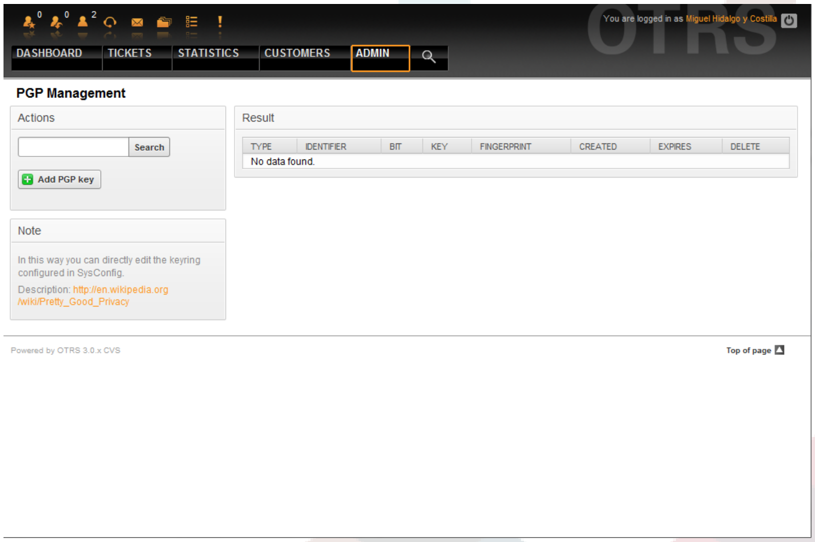
Top of page

Рисунок: Управление S/MIME.

1.10. PGP

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

Through the "PGP Keys" link on the Admin page, it is possible to manage the key ring of the user who shall be used for PGP with OTRS (see Figure below), e.g. the local OTRS user or the web server user. It is possible to add and remove keys and signatures, and you can search through all data in your key ring.



You are logged in as Miguel Hidalgo y Costilla

PGP Management

Actions

Result

TYPE	IDENTIFIER	BIT	KEY	FINGERPRINT	CREATED	EXPIRES	DELETE
No data found.							

Note

In this way you can directly edit the keyring configured in SysConfig.

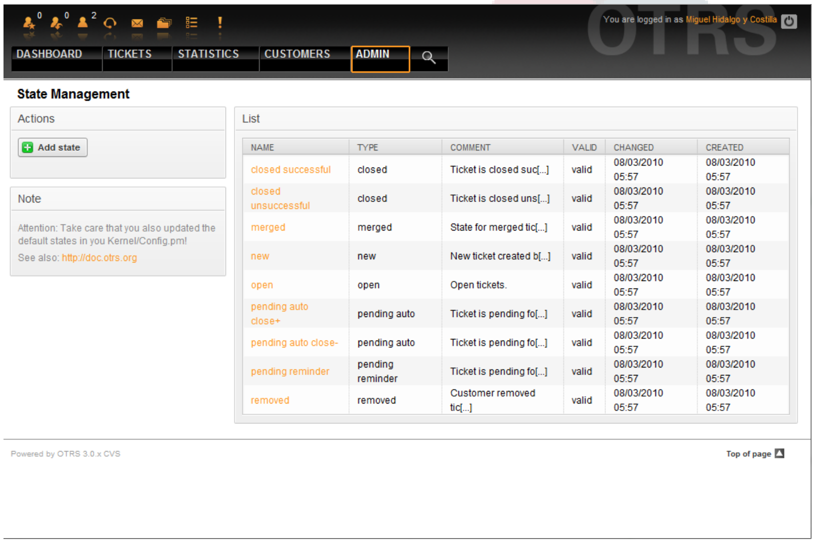
Description: http://en.wikipedia.org/wiki/Pretty_Good_Privacy

Powered by OTRS 3.0.x CVS

Рисунок: Управление PGP.

1.11. Состояния

Through the "States" link on the Admin page, you can manage the different ticket states you want to use in the system (see Figure below).



You are logged in as Miguel Hidalgo y Costilla

State Management

Actions

List

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

Note

Attention: Take care that you also updated the default states in you Kernel/Config.pm!

See also: http://doc.otrs.org

Powered by OTRS 3.0.x CVS

Рисунок: Управление состоянимями.

После установки в системе будут следующие предустановленные состояния:

- закрыта успешно
- закрыта неуспешно
- обьедененные
- новая
- открытая
- ожидание авто закрытие+
- ожидание авто закрытие-
- ожидание напоминание
- удаленная

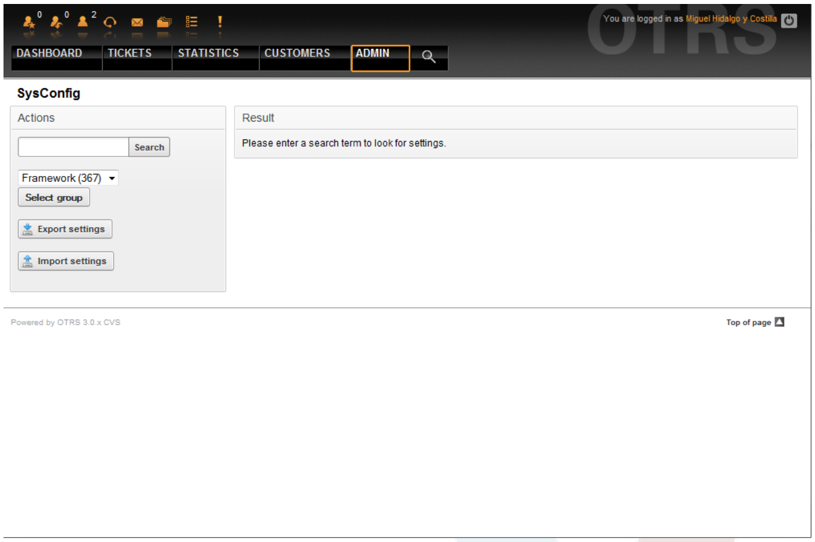
Каждое состояние связано с типом, который нужно указать когда создается новое состояние. По умолчанию есть такие типы состояний:

- закрытая
- обьедененные
- новая
- открытая
- ожидание авто
- ожидание напоминание
- удаленная

1.12. SysConfig

Ссылка SysConfig - указывает на мест,о где хранится большинство конфигурационные параметров OTRS.

The SysConfig link on the Admin page loads the graphical interface for system configuration (see Figure below). You can upload your own configuration files for the system, as well as backup all your current settings into a file. Almost all configuration parameters of the OTRS framework and installed applications can be viewed and changed through this interface. Since all configuration parameters are sorted into groups and sub groups, it is possible to navigate quickly through the vast number of existing parameters. It is also possible to perform a full-text search through all of the configuration parameters.



SysConfig

Actions

Search

Framework (367)

Select group

Export settings

Import settings

Result

Please enter a search term to look for settings.

Powered by OTRS 3.0.x CVS

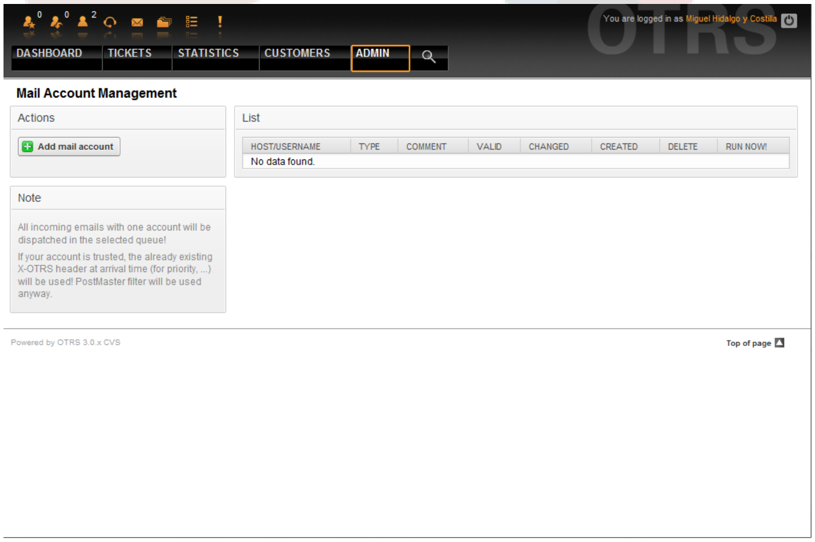
Top of page

Рисунок: Графический интерфейс для настройки системы (SysConfig).

Более подробно графический интерфейс для настройки системы рассматривается в главе "Настройка системы через веб-интерфейс" .

1.13. Использование почтовый ящиков

There are several possibilities to transport new emails into the ticket system. One way is to use a local MTA and the otrs.PostMaster.pl script that pipes the mails directly into the system. Another possibility is the use of mail accounts which can be administrated through the web interface. The "PostMaster Mail Accounts" link on the Admin page loads the management console for the mail accounts (see Figure below). OTRS supports the mail protocols: POP3, POP3S, IMAP and IMAPS.



Mail Account Management

Actions

Add mail account

List

HOST/USERNAME	TYPE	COMMENT	VALID	CHANGED	CREATED	DELETE	RUN NOW!
No data found.							

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.

Powered by OTRS 3.0.x CVS

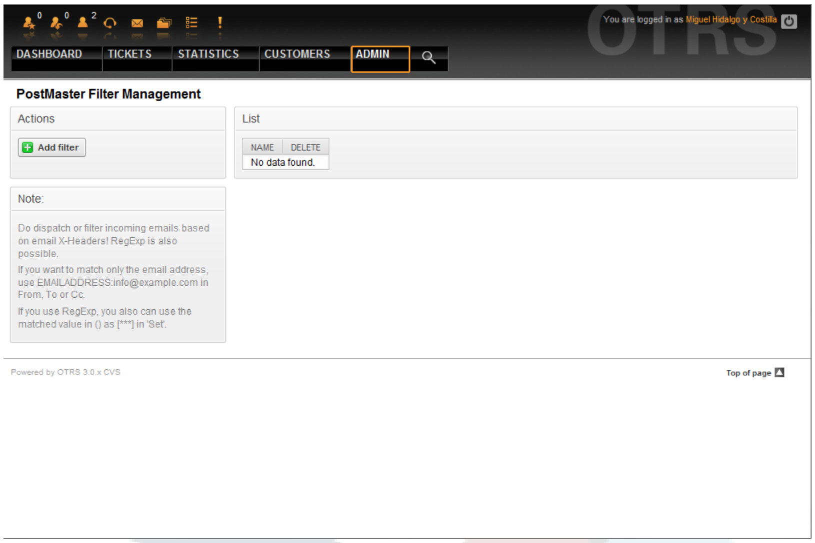
Top of page

Рисунок: Управление почтовыми ящиками.

See the section about PostMaster Mail Accounts for more details.

1.14. Filtering incoming email messages

OTRS has the capability to filter incoming email messages. For example, it is possible to put certain emails automatically into specified queues, or to set a specific state or ticket type for some mails. The filters apply to all incoming mails. You can manage your filters via the link "PostMaster Filter" on the Admin page (see Figure below).



PostMaster Filter Management

NAME	DELETE
No data found.	

Note:

Do dispatch or filter incoming emails based on email X-Headers! RegExp is also possible.

If you want to match only the email address, use EMAILADDRESS:info@example.com in From, To or Cc.

If you use RegExp, you also can use the matched value in () as [***] in 'Set'.

Powered by OTRS 3.0.x CVS

Рисунок: Управление фильтрами PostMaster.

A filter consists of one or more criteria that must be met in order for the defined actions to be executed on the email. Filter criteria may be defined for the headers or the body of an email, e.g. search for specific header entries, such as a sender address, or on strings in the body. Even regular expressions can be used for extended pattern matching. If your filter matches, you can set fields using the X-OTRS headers in the GUI. These values will be applied when creating the ticket or follow-up message in OTRS. The Table 5-4 lists the different X-OTRS headers and their meaning.

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

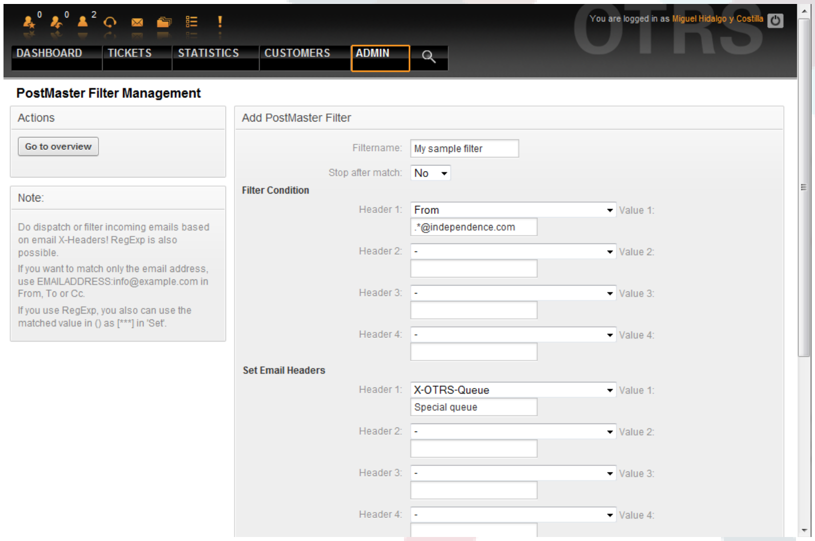
Таблица 4.5. Функции различных X-OTRS-заголовков

Название	**Возможные значения:**	**Описание**
X-OTRS-Priority:	1 очень низкий, 2 низкий, 3 нормальный, 4 высокий, 5 очень высокой	Устанавливает приоритет заявки.
X-OTRS-Queue:	Название очереди в системе.	Sets the queue where the ticket shall be sorted. If set in X-OTRS header, all other filter rules that try to sort a ticket into a specific queue are ignored. If you use a sub-queue, specify it as "Parent::Sub".
X-OTRS-Lock:	блокировать, разблокировать	Устанавливает состояние блокирования заявки.

Название	Возможные значения:	Описание
X-OTRS-Ignore:	Yes или True	Если этот X-OTRS заголовок установлен в "Yes", входящие сообщения будут полностью игнороироватся и никогда не попадут в систему.
X-OTRS-State:	новая, открытая, закрытая успешно, закрытая неуспешно, ...	Устанавливает следующее состояние заявки.
X-OTRS-State-PendingTime:	например 2010-11-20 00:00:00	Sets the pending time of a ticket (you also should sent a pending state via X-OTRS-State). You can specify absolute dates like "2010-11-20 00:00:00" or relative dates, based on the arrival time of the email. Use the form "+ $Number $Unit", where $Unit can be 's' (seconds), 'm' (minutes), 'h' (hours) or 'd' (days). Only one unit can be specified. Examples of valid settings: "+50s" (pending in 50 seconds), "+30m" (30 minutes), "+12d" (12 days). Note that settings like "+1d 12h" are not possible. You can specify "+36h" instead.
X-OTRS-Type:	по умолчанию (в зависимости от настроек)	Sets the type of a ticket (if Ticket::Type is activated).
X-OTRS-Service:	(в зависимости от настроек)	Sets the service of a ticket (if Ticket::Service is active). If you want to set a sub-service you should specify it as "Parent::Sub".
X-OTRS-SLA:	(в зависимости от настроек)	Устанавливает SLA для заявки (если активирована поддержка Ticket::Service).
X-OTRS-CustomerUser:	CustomerUser	Устанавливает клиентского пользователя для заявки.
X-OTRS-CustomerNo:	CustomerNo	Устанавливает ID пользователя для заявки.
X-OTRS-SenderType:	агент, система, клиент	Устанавливает тип отправителя заявки.
X-OTRS-ArticleType:	email-external, email-internal, email-notification-ext, email-notification-int, phone, fax, sms, webrequest,	Устанавливает тип статьи для входящих заявок.

Название	Возможные значения:	Описание
	note-internal, note-external, note-report	
X-OTRS-DynamicField-<DynamicFieldName>:	В зависимости от конфигурации Динамических Полей (Текст: Notebook, Дата: 2010-11-20 00:00:00, Количество: 1)	Saves an additional info value for the ticket on <DynamicFieldName> Dynamic Field.
X-OTRS-Loop:	True	Если этот X-OTRS заголовок установлен, то отправителю сообщение недоставляется ни одного автоматического ответа (защита от зацыкливания).

You should specify a name for every filter rule. Filter criteria can be specified in the section "Filter Condition". Choose via the listboxes for "Header 1", "Header 2" and so on for the parts of the messages where you would like to search, and specify on the right side the values you wish to filter on. In the section "Set Email Headers", you can choose the actions that are triggered if the filter rules match. You can select for "Header 1", "Header 2" and so on to select the X-OTRS-Header and set the associated values (see Figure below). Filter rules are evaluated in alphabetical order, and are all executed except if the "Stop after match" setting has been set to "Yes" in one of the rule (in this case evaluation of the remaining filters is cancelled).



PostMaster Filter Management

Actions

Note:

Do dispatch or filter incoming emails based on email X-Headers! RegExp is also possible.

If you want to match only the email address, use EMAILADDRESS:info@example.com in From, To or Cc.

If you use RegExp, you also can use the matched value in () as [***] in 'Set'.

Add PostMaster Filter

Filtername: My sample filter

Stop after match: No

Filter Condition

Header 1: From Value 1: .*@independence.com

Header 2: - Value 2:

Header 3: - Value 3:

Header 4: - Value 4:

Set Email Headers

Header 1: X-OTRS-Queue Value 1: Special queue

Header 2: - Value 2:

Header 3: - Value 3:

Header 4: - Value 4:

Рисунок: Добавление фильтров PostMaster.

Пример 4.1. Сортирование спама в определенную очередь

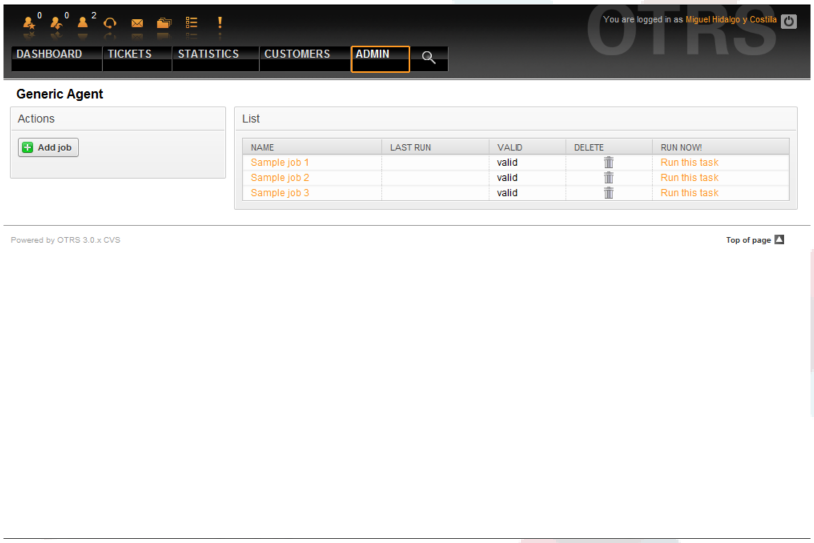
A useful filter rule would be to let OTRS automatically move mails marked for spam ,by using a spam detection tool such as SpamAssassin, into the "Junk" queue. SpamAssassin adds the "X-Spam-Flag" header to every checked mail. When the mail is marked as spam, the Header is set to "Yes". So the filter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

There are additional modules, that can be used to filter incoming messages more specifically. These modules might be useful when dealing with larger, more complex systems.

1.15. Выполнение автоматизированных заданий с помощью GenericAgent

The GenericAgent is a tool to execute tasks automatically. The GenericAgent, for example, can close or move tickets, send notifications on escalated tickets, etc.

Click the link "GenericAgent" on the Admin page (see Figure below). A table with all automated jobs in the system is displayed. These jobs can then be edited, run manually or removed entirely.



Generic Agent

Actions

Add job

List

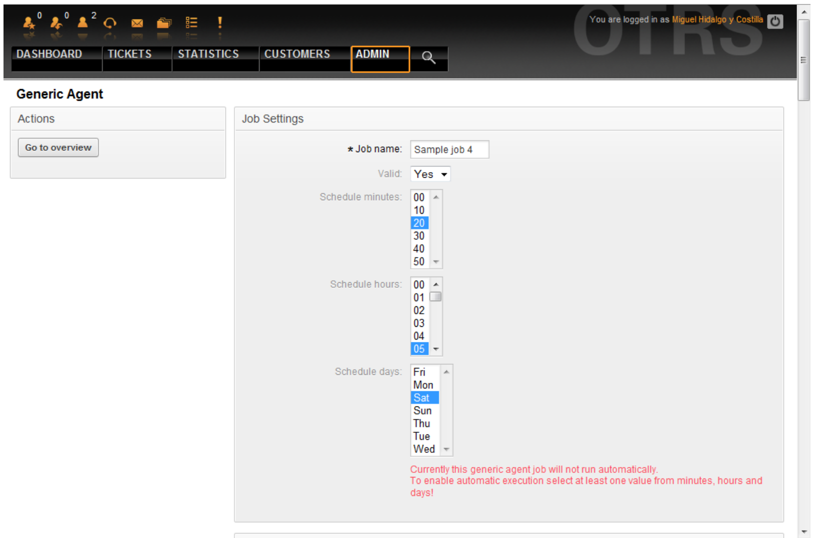
NAME	LAST RUN	VALID	DELETE	RUN NOW!
Sample job 1		valid	🗑	Run this task
Sample job 2		valid	🗑	Run this task
Sample job 3		valid	🗑	Run this task

Powered by OTRS 3.0.x CVS

Top of page

Рисунок: Список заданий для GenericAgent.

Click the "Add job" button to create a new job. You first need to supply a name. Then you can specify how the job will be executed: automatic at fixed times (like a cronjob, this mode will operate on all tickets found by the ticket filter) or based on ticket events (right after a single ticket was modified, if it matches the ticket filter). Note that if you manually run event based jobs from the overview screen, they will operate on all tickets fund by the ticket filter.



You are logged in as Miguel Hidalgo y Costilla

Generic Agent

Job Settings

* Job name: Sample job 4

Valid: Yes

Schedule minutes: 00 10 20 30 40 50 (selected: 20)

Schedule hours: 00 01 02 03 04 05 (selected: 05)

Schedule days: Fri Mon Sat Sun Thu Tue Wed (selected: Sat)

Currently this generic agent job will not run automatically.
To enable automatic execution select at least one value from minutes, hours and days!

Рисунок: Создание заданий для GenericAgent.

For every job, you can specify a ticket filter, for example to only operate on tickets in a certain queue. All filter criteria must be met for a job to be run on a ticket.

Finally, the ticket can be modified by setting various ticket fields like a new queue or state. It is possible to attach a note to the ticket(s) or run a execute a customized module. You also have the option to delete the ticket(s) from the database. This can be useful to purge outdated or invalid data from the system.

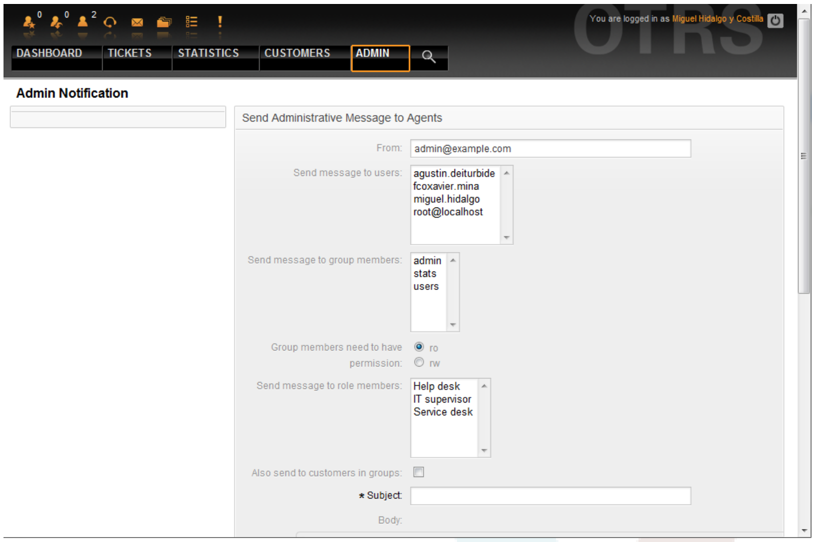
Предупреждение

If you use the ticket delete function, all affected tickets and their attachments will be removed from the database and cannot be restored!

After editing a job, OTRS will return to the overview screen. There you have the possibility to run any job manually. If you choose to run a job, you will first see all tickets which will be affected when the job actually is run. This list helps you to verify that the job is working as intended. At this point no changes have been made to these tickets yet. Only if you confirm the screen the job will be executed.

1.16. Эллектронная почта администратора

OTRS administrators can send messages to specific users or groups. The "Admin Notification" link on the Admin page opens the screen where the agents and groups that should be notified can be selected (see Figure below).



You are logged in as Miguel Hidalgo y Costilla

Admin Notification

Send Administrative Message to Agents

From: admin@example.com

Send message to users: agustin.deiturbide
fcoxavier.mina
miguel.hidalgo
root@localhost

Send message to group members: admin
stats
users

Group members need to have permission: ● ro ○ rw

Send message to role members: Help desk
IT supervisor
Service desk

Also send to customers in groups: □

* Subject:

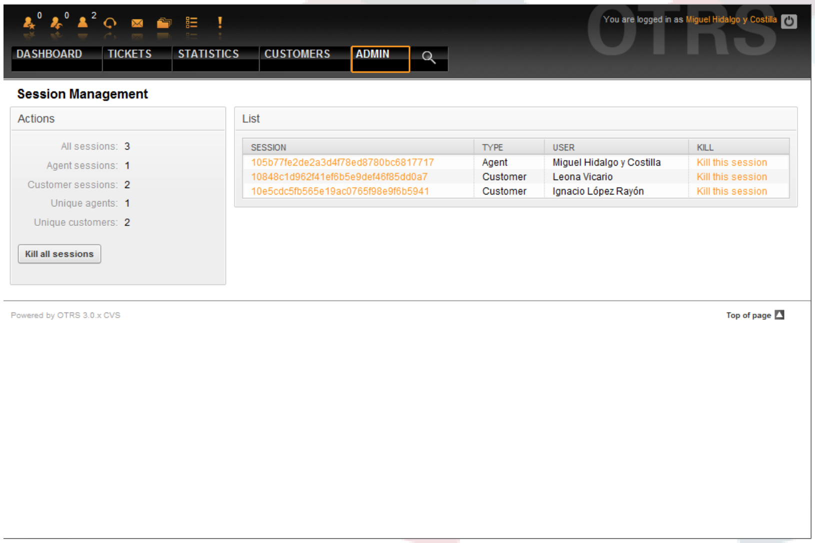
Body:

Рисунок:. Оповещения Администратором.

Можно указать отправителя, тему и текст уведомления. Также можно выбрать агентов, группы и роли, которые должны получить сообщение.

1.17. Управление сеансами

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



You are logged in as Miguel Hidalgo y Costilla

Session Management

Actions

All sessions: 3
Agent sessions: 1
Customer sessions: 2
Unique agents: 1
Unique customers: 2

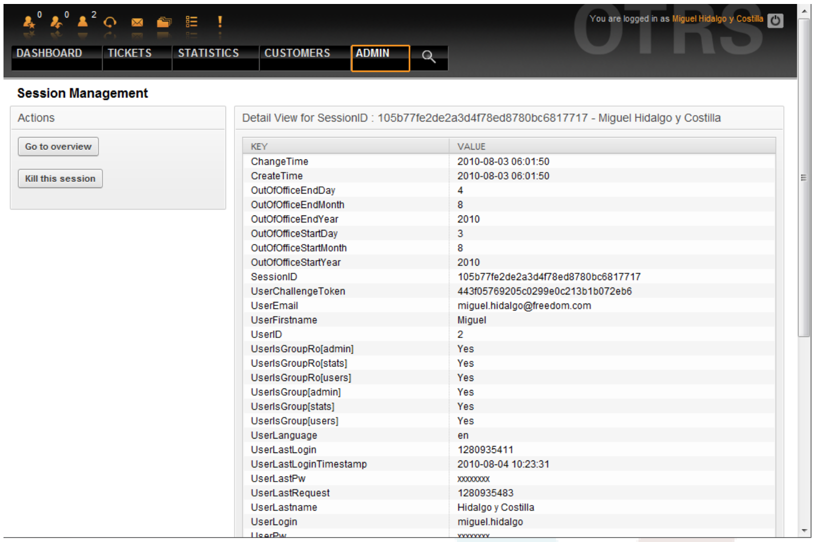
List

SESSION	TYPE	USER	KILL
105b77fe2de2a3d4f78ed8780bc6817717	Agent	Miguel Hidalgo y Costilla	Kill this session
10848c1d962f41ef6b5e9def46f85dd0a7	Customer	Leona Vicario	Kill this session
10e5cdc5fb565e19ac0765f98e9f6b5941	Customer	Ignacio López Rayón	Kill this session

Powered by OTRS 3.0.x CVS

Рисунок: Управление Сеансами.

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



You are logged in as Miguel Hidalgo y Costilla

Session Management

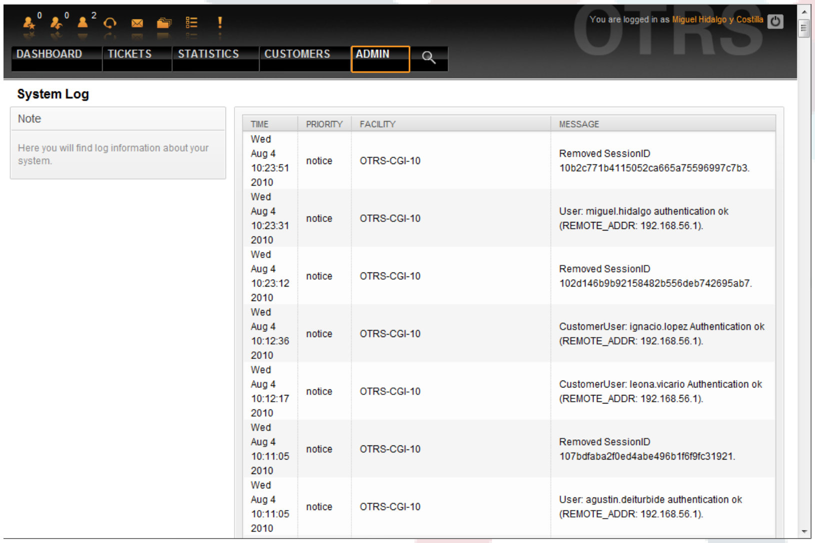
Detail View for SessionID : 105b77fe2de2a3d4f78ed8780bc6817717 - Miguel Hidalgo y Costilla

KEY	VALUE
ChangeTime	2010-08-03 06:01:50
CreateTime	2010-08-03 06:01:50
OutOfOfficeEndDay	4
OutOfOfficeEndMonth	8
OutOfOfficeEndYear	2010
OutOfOfficeStartDay	3
OutOfOfficeStartMonth	8
OutOfOfficeStartYear	2010
SessionID	105b77fe2de2a3d4f78ed8780bc6817717
UserChallengeToken	443f05769205c0299e0c213b1b072eb6
UserEmail	miguel.hidalgo@freedom.com
UserFirstname	Miguel
UserID	2
UserIsGroupRo[admin]	Yes
UserIsGroupRo[stats]	Yes
UserIsGroupRo[users]	Yes
UserIsGroup[admin]	Yes
UserIsGroup[stats]	Yes
UserIsGroup[users]	Yes
UserLanguage	en
UserLastLogin	1280935411
UserLastLoginTimestamp	2010-08-04 10:23:31
UserLastPw	xxxxxxxx
UserLastRequest	1280935483
UserLastname	Hidalgo y Costilla
UserLogin	miguel.hidalgo
UserPw	xxxxxxxx

Рисунок: Подробная информация о сессии.

1.18. Системный журнал

The "System Log" link on the Admin page shows the log entries of the system, reverse chronologically sorted with most recent first (see Figure below).



You are logged in as Miguel Hidalgo y Costilla

System Log

Here you will find log information about your system.

TIME	PRIORITY	FACILITY	MESSAGE
Wed Aug 4 10:23:51 2010	notice	OTRS-CGI-10	Removed SessionID 10b2c771b4115052ca665a75596997c7b3.
Wed Aug 4 10:23:31 2010	notice	OTRS-CGI-10	User: miguel.hidalgo authentication ok (REMOTE_ADDR: 192.168.56.1).
Wed Aug 4 10:23:12 2010	notice	OTRS-CGI-10	Removed SessionID 102d146b9b92158482b556deb742695ab7.
Wed Aug 4 10:12:36 2010	notice	OTRS-CGI-10	CustomerUser: ignacio.lopez Authentication ok (REMOTE_ADDR: 192.168.56.1).
Wed Aug 4 10:12:17 2010	notice	OTRS-CGI-10	CustomerUser: leona.vicario Authentication ok (REMOTE_ADDR: 192.168.56.1).
Wed Aug 4 10:11:05 2010	notice	OTRS-CGI-10	Removed SessionID 107bdfaba2f0ed4abe496b1f6f9fc31921.
Wed Aug 4 10:11:05 2010	notice	OTRS-CGI-10	User: agustin.deiturbide authentication ok (REMOTE_ADDR: 192.168.56.1).

Рисунок: Системный журнал.

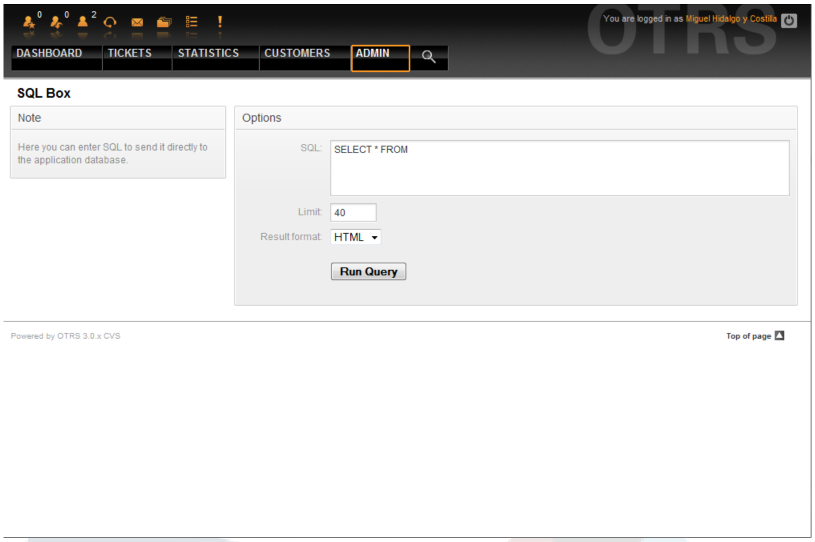
Каждая строка в журнале содержит: время, приоритет, компонент системы и непосредственно саму запись.

Примечание

System logs are available via the web interface only on Linux / Unix systems. On Windows systems, you can see the logs using a text editor by opening the file `[install_dir]otrs\var\log\otrs.log`.

1.19. SQL запросы

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


You are logged in as Miguel Hidalgo y Costilla
DASHBOARD
TICKETS
STATISTICS
CUSTOMERS
ADMIN
Run Query
Powered by OTRS 3.0.x CVS
Top of page

SQL Box

Note

Here you can enter SQL to send it directly to the application database.

Options

SQL: SELECT * FROM

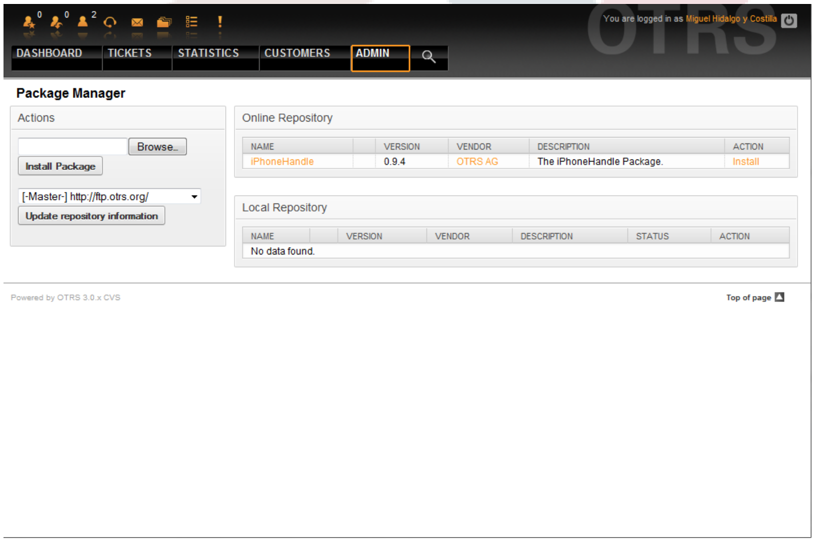
Limit: 40

Result format: HTML

Рисунок: SQL-запросы.

1.20. Менеджер пакетов

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


You are logged in as Miguel Hidalgo y Costilla
DASHBOARD
TICKETS
STATISTICS
CUSTOMERS
ADMIN
Browse..
Install Package
Update repository information
Powered by OTRS 3.0.x CVS
Top of page

Package Manager

Actions

[-Master-] http://ftp.otrs.org/

Online Repository

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

Local Repository

NAME	VERSION	VENDOR	DESCRIPTION	STATUS	ACTION
No data found.					

Рисунок: Менеджер пакетов.

Менеджер Пакетов позволяет просмотреть пакеты расширения OTRS вместе с номерами их версий, установленные на сервере в данный момент.

Пакеты можно установить из удаленного хоста, для этого, в разделе *Онлайн Репозиторий* нужно выбрать репозиторий и нажать кнопку *Обновить информацию из репози-*

тория. Все доступные пакеты будут отображены в соответствующей таблице. В правой части страницы отображаются доступные пакеты. Чтобы установить пакет, нажмите на ссылку *Установить*. После установки пакет будет отображаться в разделе *Local Repository*.

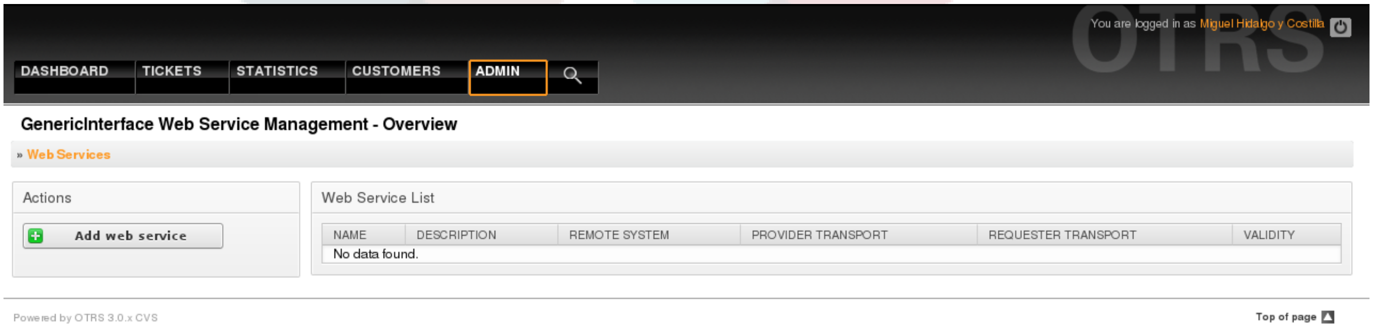
To upgrade an installed package, the list of available packages in the online repository will show *Upgrade* in the Action column for any package that has a higher version than the one that is installed locally. Just click Upgrade and it will install the new package version on your system.

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

В особых случаях, можно настроить Менеджер Пакетов, например, для использования прокси-сервера, или локального репозитория. перейдя по ссылке Framework:Core::Package, просто посмотрите на имеющиеся варианты в файле SysConfig.

1.21. Веб-сервисы

The Web Services link leads to the graphical interface where web services (for the OTRS Generic Interface) are created and maintained (see Figure below).



GenericInterface Web Service Management - Overview

Web Service List

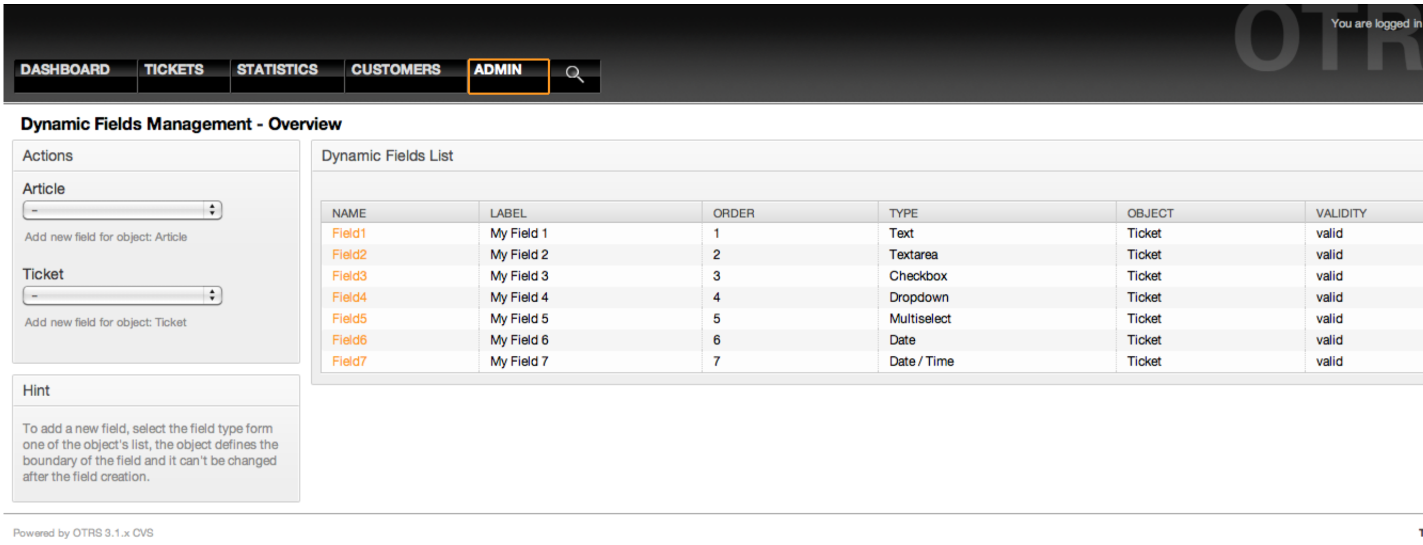
NAME	DESCRIPTION	REMOTE SYSTEM	PROVIDER TRANSPORT	REQUESTER TRANSPORT	VALIDITY
No data found.					

Рисунок: Графический интерфейс для веб-сервисов.

The graphical interface for web services configuration is described in more detail in the section "Web Service Graphical Interface".

1.22. Динамические Поля

Dynamic Fields is the place where you setup and manage custom fields for tickets and articles (see figure below).



Dynamic Fields Management - Overview

Actions

Article

-

Add new field for object: Article

Ticket

-

Add new field for object: Ticket

Hint

To add a new field, select the field type form one of the object's list, the object defines the boundary of the field and it can't be changed after the field creation.

Dynamic Fields List

NAME	LABEL	ORDER	TYPE	OBJECT	VALIDITY
Field1	My Field 1	1	Text	Ticket	valid
Field2	My Field 2	2	Textarea	Ticket	valid
Field3	My Field 3	3	Checkbox	Ticket	valid
Field4	My Field 4	4	Dropdown	Ticket	valid
Field5	My Field 5	5	Multiselect	Ticket	valid
Field6	My Field 6	6	Date	Ticket	valid
Field7	My Field 7	7	Date / Time	Ticket	valid

Powered by OTRS 3.1.x CVS

Рисунок: Страница с некоторыми динамическими полями.

The dynamic fields configuration is described in more detail in the section "Dynamic Fields Configuration".

Each dynamic field type has its own configuration settings and therefore its own configuration screen.

Примечание

In the OTRS framework, dynamic fields can only be linked to tickets and articles by default, but they can be extended to other objects as well.

2. Настройка Системы

2.1. Конфигурационные файлы OTRS

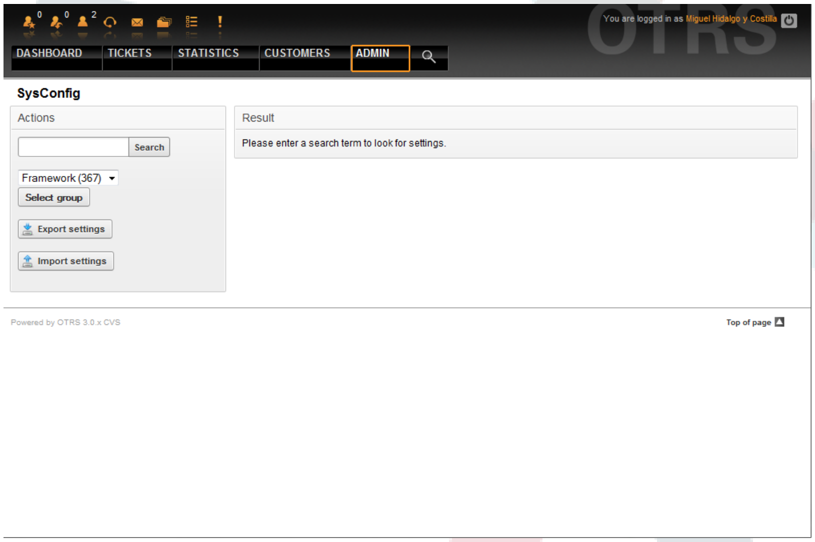
Все конфигурационные файлы OTRS хранятся в директории `Kernel` и ее поддиректориях. Нет необходимости изменять другие файлы кроме `Kernel/Config.pm`, потому что остальные файлы будут изменены, после обновления системы. Просто скопируйте конфигурационные параметры из других файлов в `Kernel/Config.pm` и измените их в соответствии с вашими потребностями. Этот файл никогда не будет затронут во время обновления, так что настройки, которые вы произвели вручную будут сохранены.

В каталоге `Kernel/Config/Files` есть некоторые другие файлы, которые передаются, когда запускается страница входа в OTRS. Если установлены дополнительные приложения, такие как FAQ (ЧАВО) или Менеджер Файлов, то их конфигурационные файлы также могут быть найдеты по указанному пути.

If the OTRS web interface 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 files `Kernel/Config/Files/ZZZAAuto.pm`, `Kernel/Config/Files/ZZZAuto.pm` and `Kernel/Config/Files/ZZZProcessManagement.pm` (if it exists) will be evaluated. These files are used by the graphical interface for system configuration caching 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.

2.2. Настройка системы через графический веб-интерфейс

Since OTRS 2.0, nearly all configuration parameters of the central framework or additional installed applications, can be changed easily with the graphical interface for system configuration. Log in as OTRS administrator and follow the SysConfig link on the Admin page to execute the new configuration tool (see Figure below).



You are logged in as Miguel Hidalgo y Costilla

SysConfig

Please enter a search term to look for settings.

Powered by OTRS 3.0.x CVS

Рисунок: Графический интерфейс для настройки системы.

Поскольку в настоящее время в OTRS имеется больше 600 параметров конфигурации, есть несколько различных способов быстрого доступа к специфическим настройкам. Используя полнотекстовый поиск можно просмотреть все конфигурационные параметры на предмет совпадения с определенным ключевым словом. Полнотекстовый поиск ищет совпадения не только в названиях конфигурационных параметров, но также и в их описаниях. Это позволяет найти тот или инной элемент даже в том случае если его название неизвесно.

Кроме этого, все параметры конфигурации отсортированы по основным группам и подгруппам. Главная группа представляет собой приложение соостоящие из подгрупп "Framework" для главного OTRS-фреймворка, "Ticket" для системы обработки заявок, "FAQ" - для системы ответов и вопросов и так далее. Подгруппы могут быть доступны если приложение выбирается из списка групп "Выборор групп" кнопка нажата.

Каждый параметр конфигурации может быть "включен" или "выключен" с помощью флажка. Если параметр "выключен" (флажок не выбран) система будет игнорировать этот параметр или использовать настройки по умолчанию. Также, воспользовавшиь кнопкой "Восстановить" можно переключится между измененным значением конфигурационного параметра и его значением по умолчанию.

Если нужно сохранить все изменения, внесенные в конфигурацию системы, например, для быстрой настройки новой установки, можно воспользоваться кнопкой "Экспортировать настройки", которая создаст .pm-файл. Чтобы восстановиь свои собственные настройки, нажмите кнопку "Импортировать настройки" и выберите .pm-файл с настройками созданными ранее.

Примечание

Из за соображений безопасности, параметры конфигурации соединения с базой даных не могут быть изменены в SysConfig. Их нужно установлены вручную в файле `Kernel/Config.pm`.

3. Резервное копирование системы

В этой главе описывается резервное копирование и восстановление данных OTRS.

3.1. Резервное копирование

Есть два типа данных для резервного копирования: файлы приложения (например, файлы в `/opt/otrs`) и данные, хранящиеся в базе данных.

To simplify backups, the script `scripts/backup.pl` is included with every OTRS installation. It can be run to backup all important data (see Script below).

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

Сценарий: Получене помощи о механизме резервного копированиия OTRS.

Для создания резервной копии выполните команду, приведенную в Сценарии ниже:

```
linux:/opt/otrs/scripts# ./backup.pl -d /backup/
Backup /backup//2010-09-07_14-28/Config.tar.gz ... done
Backup /backup//2010-09-07_14-28/Application.tar.gz ... done
Dump MySQL rdbms ... done
Compress SQL-file... done
linux:/opt/otrs/scripts#
```

Сценарий:Создание резервной копии.

Все данные хранятся в директории `/backup/2010-09-07_14-28/` (см. Сценарий ниже). Кроме того данные сохранены в файле с расширением .tar.gz.

```
linux:/opt/otrs/scripts# ls /backup/2010-09-07_14-28/
Application.tar.gz  Config.tar.gz  DatabaseBackup.sql.gz
linux:/opt/otrs/scripts#
```

Сценарий: Проверка файлов резервной копии.

3.2. Восстановление

Для восстановления резервной копии, сохраненные данные приложения должны быть записаны обратно в каталог установки, например `/opt/otrs`. База данных также должна быть восстановлена.

A script `scripts/restore.pl` (see Script below), which simplifies 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#
```

Сценарий: Получение справки о механизме восстановления.

Data that is stored, for example, in the directory `/backup/2010-09-07_14-28/`, can be restored with the command specified in the script below, assuming the OTRS installation is at `/opt/otrs`.

```
linux:/opt/otrs/scripts# ./restore.pl -b /backup/2010-09-07_14-28 -d /opt/otrs/
Restore /backup/2010-09-07_14-28//Config.tar.gz ...
Restore /backup/2010-09-07_14-28//Application.tar.gz ...
create MySQL
decompresses SQL-file ...
cat SQL-file into MySQL database
compress SQL-file...
linux:/opt/otrs/scripts#
```

Сценарий: Восстановление данных OTRS.

4. Настройки электронной почты

4.1. Отправка / получение электронной почты

4.1.1. Отправка почты

4.1.1.1. через Sendmail (по умолчанию)

OTRS может отправлять сообщения электронной почты через Sendmail, Postfix, Qmail or Exim). По умолчанию используется Sendmail, он должен работать при установки системы "из коробки".

Параметры sendmail можно настроить через графический веб-интерфейс для конфигурации (Framework::Core::Sendmail)

4.1.1.2. Через SMTP-сервер или smarthost

OTRS может отсылать сообщения электронной почты через SMTP (Simple Mail Transfer Protocol / RFC 821) или Secure SMTP. Возможно вы захотите использовать эту опцию на не-unix-системах (например на Windows).

Параметры настройки SMTP-сервера могут быть сконфигурированны через SysConfig (Framework::Core::Sendmail). Если у вас нету SMTPS в качестве опции, значит вы пропустили установку соответствующих Perl-модулей. В таком случае, обратитесь пожалуйста к этой "Установка Perl-модулей необходимых для работы OTRS" инструкции.

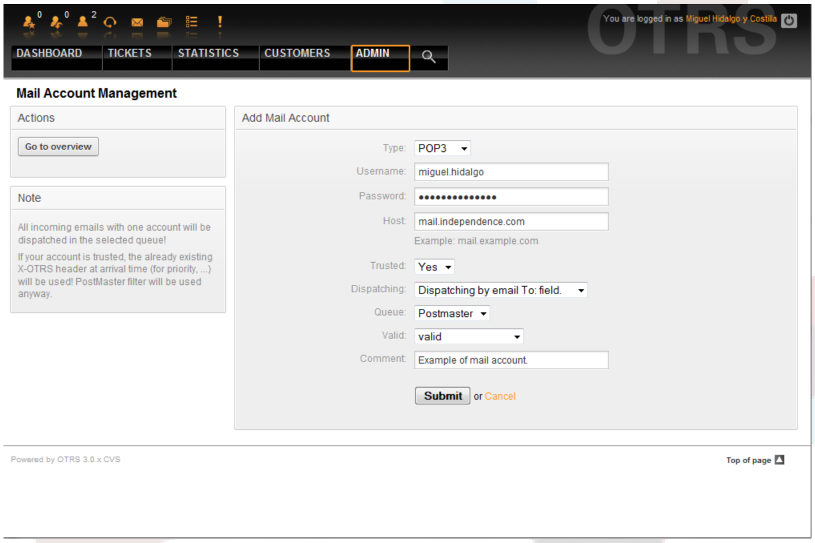
4.1.2. Получение сообщений электронной почты

4.1.2.1. Учетные записи электронной почты настраиваемые через графический пользовательский интерфейс OTRS

OTRS позволяет получать сообщения электронной почты через почтовые аккаунты POP3, POP3S, IMAP, и IMAPS.

Для настройки почтовых аккаунтов воспользуйтесь ссылкой "Почтовые Аккаунты PostMaster" на странице администрирования.

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



Mail Account Management

Actions

Go to overview

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.

Add Mail Account

Type: POP3
Username: miguel.hidalgo
Password: ••••••••••••••
Host: mail.independence.com
Example: mail.example.com
Trusted: Yes
Dispatching: Dispatching by email To: field.
Queue: Postmaster
Valid: valid
Comment: Example of mail account.

Submit or Cancel

Powered by OTRS 3.0.x CVS

Top of page

Рисунок Добавление учетной записи электронной почты.

Если для опции "Trusted" выбрано значение "Да", будут оцениваться и выполнятся любые X-OTRS-заголовки, присоединенные к входящему сообщению. Поскольку X-OTRS-заголовок может выполнять некоторые действия в системе обработки заявок, то для извесных отправителей опцию Trusted нужно установить только в значение Да. X-OTRS-заголовки используюся в OTRS модулем фильтрации . Более подробно X-OTRS заголовки рассматриваются в этой таблице. Любые созданные и выполненные правила фильтрации независимы от параметров настроки Trusted.

Можно контролировать распределением входящих сообщений, если они должны сортироватся по очередях или содержимому поля "Кому". Если для Диспетчеризации выбрана опция "Диспетчеризация по выбранной очереди", то все входящие сообщения будут отсортированы в указанной очереди. В таком случае адресс, с которого отправили сообщение не учитывается. Если для диспетчеризации выбрана опция "Диспетчеризация емейлов по полю Кому", система для входящих емейлов проверяет, связана ли эта очередь с адресом электронной почты в поле Кому. Перейдя по ссылке Управление E-mail-адресами в Панели Администрирования, можно связать определенный адрес с очередью. Если не найдено никакой связи между адресом в поле "Кому" то сообщение будет сохранятся в системе в "Raw" очереди, которая есть PostmasterDefaultQueue по умолчанию после установки системы.

Все данные для учетных записей электронной почты сохраняются в базе данных OTRS. Скрипт `otrs.PostMasterMailbox.pl`, который находится в директории bin вашей системы OTRS, использует настройки в базе данных и получает почту. Вы можете выполнить файл `./bin/otrs.PostMasterMailbox.pl` вручную чтобы проверить что все ваши настройки почты работают правильно.

Для обычной установки, с помощью файла заданий cron job `postmaster_mailbox` выборка емейлов будет выполнятся каждые 10 минут. Для получения дополнительной информации об изменении заданий планировщика задач cron jobs, обратитесь к главе "Настройка пралировщика задач cron jobs для OTRS".

Примечание

При получении почты OTRS удаляет почту с POP или IMAP сервера. Нету такой опции, которая бы позволяла хранить копию сообщения на сервере. Если вы все же хотите чтобы такая возможность присутствовала, скорей всего нужно воспользоваться правилами переадресации на почтовый сервер. Оратитесь пожалуйста к документации по вашему почтовому серверу.

4.1.2.2. Через командную строку программы и например, procmail (otrs.PostMaster.pl)

Если для получения электронной почты в OTRS нету возможности использовать учетные записи, то эту проблему можно решить с помощью программы командной строки `bin/otrs.PostMaster.pl`Она принимает сообщения электронной почты через STDIN и направляет их непосредственно в OTRS. Это значит что емейлы будут доступны в OTRS когда MDA (Mail Delivery Agent - Агент доставки почты) выполняет эту программу.

Для тестирования `bin/otrs.PostMaster.pl` без MDA, выполните команду приведенную в листинге нижеприведенного сценария.

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

Сценарий: Тестирование PostMaster без MDA.

Если сообщения электронной почты отображаются в QueueView, значит вашы настройки работают.

Procmail - это очень распостраненный фильтр электронной почты в среде Linux. Он устанавливается на большинстве систем. Если нет, перейдите по ссылке *procmail homepage*.

Для настройки procmail для OTRS (требуется сконфигурированный транспортный агент MTA, например sendmail, postfix, exim or qmail), используйте файл `~otrs/.procmailrc.dist`, скопируйте его в `.procmailrc` а затем добавьте строки из нижеприведенного сценария.

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

```
| $SYS_HOME/bin/otrs.PostMaster.pl
```

Сценарий: Настройка procmail для OTRS.

Все емейлы отсылаемые локальному OTRS-пользователю будут обрабатываться `bin/otrs.PostMaster.pl` и потом отображаться в QueueView.

4.1.2.3. Получение электронной почты по протоколу POP3 или IMAP и обработка для otrs.PostMaster.pl

Для того чтобы получить электронную почту с вашего почтового сервера через POP3 или IMAP и сохранить ее на компьютере на котором установлен OTRS, для локального аккаунта или в procmail, перейдите по ссылке fetchmail.

Примечание

Работающий и сконфигурированный SMTP необходим для работы OTRS.

Можно использовать файл `.fetchmailrc.dist` в домашней директории OTRS и скопировать его в `.fetchmailrc`. Изменить его в соответствии с вашими требованиями (см. ниже Пример 7-1).

Пример 4.2. .fetchmailrc

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

Не забудьте установить.fetchmailrc to 710 ("chmod 710 .fetchmailrc")!

Из Листинга 7-1 выше, `.fetchmailrc` , все емейлы будут перенаправлены в локальный OTRS-аккаунт, если выполнена команда **fetchmail -a**. Установите эту команду в планировщике задач cronjob если хотите извлекать емейлы постоянно.

4.1.2.4. Фильтрация/рассылка модулями OTRS/PostMaster (для более сложной диспетчеризации)

Если вы используете метод bin/otrs.PostMaster.pl или bin/otrs.PostMasterMailbox.pl, то с модулем фильтрации PostMaster можно вставить или модифицировать X-OTRS заголовок. С помощью X-OTRS-заголовков, система обработки заявок может вызывать некоторые действия для входящих сообщений, сортировать их в определенные очереди, или, например, изменять приоритет или ID-клиента. Более подробную информацию о X-OTRS-заголовках можно найти в главе добавление аккаунтов электронной почты в Панели Администрирования.

Есть некоторые предустановленные модули фильтрации:

Примечание

Название задания (например $Self->{'PostMaster::PreFilterModule'}->{'JobName'}) должно быть уникальным!

Kernel::System::PostMaster::Filter::Match модуль по умолчанию для проверки совпадения заголовков определенных емейлов (например "От", "Кому", "Тема", ...). Он может устанавливать новые заголовки email (например X-OTRS-Ignore: да или X-OTRS-Queue:

spam) если совпадают правила совпадения. Задания из Примера 7-2 могут быть прописаны в `Kernel/Config.pm`

Пример 4.3. Пример задания для модуля фильтрации 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 модуль по умолчанию для получения емейлов для внешних команд. Вывод передается в STDOUT и если результат истинна, то устанавливается новый заголовок (например X-OTRS-Ignore: да или X-OTRS-Queue: spam). Пример 7-3 может быть использован в `Kernel/Config.pm`

Пример 4.4. Пример задания для модуля фильтрации 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',
    },
};
```

Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition is a default module that adds the possibility to parse external identifiers, in the email subject, the body or both using regular expressions. It then stores this value in a defined dynamic field. When an email comes in, OTRS will first search for an external identifier and when it finds one, query OTRS on the pre-defined dynamic field field. If it finds an existing ticket, it will update this ticket, otherwise it will create a new ticket with the external reference number in the separate field.

OTRS SysConfig already provide 4 different settings to setup different external ticket numbers. If more settings are needed they need to need to be added manually. The following example can be used in `Kernel/Config.pm` to extend SysConfig settings.

Пример 4.5. Example job for the filter module Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition

```
   # Job Name: ExternalNumber
   # External Ticket Number Reconition, check for Incident-<number> in incoming mails subject
and
   # body from the addeesses <sender>@externalticket.com, if number is found it will be stored
in
   # the dynamic field 'ExternalNumber' (that need to be setup in the Admin Panel).
   $Self->{'PostMaster::PreFilterModule'}->{'000-ExternalTicketNumberRecognition5'} =  {
       'FromAddressRegExp' => '\\s*@externalticket.com',
       'NumberRegExp'      => 'Incident-(\\d.*)',
       'SearchInSubject'   => '1',
       'SearchInBody'      => '1',
       'TicketStateTypes'  => 'new;open'
       'DynamicFieldName'  => 'ExternalNumber',
       'Module'            =>
'Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition',
       'Name'              => 'Test External Ticket Number',
       'SenderType'        => 'system',
   };
```

Configuration Options

- FromAddressRegExp

  This is an optional setting. Only mails matching this "From:" address will be considered for this filter. You can adjust this setting to the sender address your external system uses for outgoing mails. In case this address can differ, you can set this option to empty. OTRS will in that case not check the sender address.

- NumberRegExp

  This is a mandatory setting. This setting contains the regular expression OTRS will use to extract the ticket number out of the subject and/or ticket body. The default regular expression will match occurrences of for example 'Incident-12354' and will put the part between parentheses in the dynamic field field, in this case '12354'.

- SearchInSubject

  If this is set to '1', the email subject is searched for a ticket number.

- SearchInBody

  If this is set to '1', the email body is searched for a ticket number.

- TicketStateTypes

  This is an optional setting. If given, it will search OTRS only for open external tickets of given state types. The state types are separated with semicolons.

- DynamicField

  This is a required setting. It defines the dynamic filed that is used to store the external number (the field name must exist in the system and has to be valid).

- SenderType

  This defines the sender type used for the articles created in OTRS.

Конечно, также есть возможность разработки своих собственных PostMaster-модулей фильтрации.

4.2. Безопасность электронной почты с PGP

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

1. Установка GnuPG с помощью менеджера пакетов вашей операционной системы.

2. Настройте GnuPG для использования с OTRS. Для GnuPG и приватного ключа нужно создать необходимые директории. В командной строки для пользователя 'otrs' нужно выполнить команду из нижеприведенного сценирия.

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

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

You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

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

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

```
Passphrase: secret
Repeat passphrase: secret

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

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

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

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

linux:~$
```

Сценарий: Настройка GnuPG.

Как показано в нижеприведенном сценарии, настройки по умолчанию могут быть применены для большинства из необходимых параметров. Правильно должны быть указаны только пароль и значения для владельца ключа.

3. Теперь OTRS готов для использования PGP. Через Панель Администрирования откройте интерфейс SysConfig и найдите ссылку "PGP". Из результатов поиска выберите подгруппу Crypt::PGP.

   На странице с PGP-настройками, PGP должен быть активирован для OTRS (первая опция). Также нужно установить ипроверить путь к gpg-программе.

   Следующим нужно изменить конфигурационный параметр (PGP::Options). Используя эти конфигурационные настройки можно определить параметры, которые будут использоваться для каждого вызова gpg пользователем 'otrs'. Каталог с конфигурационными файлами для GnuPG является очень важным. В примере используется `/opt/otrs/.gnupg`. Эта директоиря была создана на первом шаге конфигурации PGP.

   Используя следующий конфигурационный параметр (PGP::Key::Password), можно указать пары для ключей ID и их пароли для собственных закрытых ключей. Поскольку партнеры извне пишут зашифрованные сообщения в систему заявок используя ваш публичный ключ, OTRS может расшифровать эти сообщения с помощью ID/паролей указаных здесь.

   Как получить идентификатор вашего собственного закрытого ключа? ID вашего собственного закрытого идентификатора отображается во время генерации (см. выше шаг 1). Также есть возможность получить ID, указав команду из нижеприведенного сценария вызванную с правами пользователя 'otrs':

```
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:~$
```

Сценарий: Получение ID вашего собственного приватного ключа.

Идентификатор (ID) закрытого ключа можно найти в строке, которая начинается с "sub". Это шестнадцатеричное представление строки, длиной в восемь символов, например "52B97069". Пароль, который нужно указать для этого ключа в системе обработки заявок такой же как данные при генерации ключей.

После того, как эти данные были введены, нажмите кнопку "Обновить" чтобы сохранить настройки. Теперь OTRS готов принимать и расшифровывать зашифрованные сообщения.

4. И наконец, нужно импортировать открытый ключ клиента. Это гарантирует, что зашифрованные сообщения могут быть отправлены этому клиенту. Есть два способа импортировать открытый ключ клиента.

   Первый способ - указать открытый ключ клиента в веб-интерфейсе управления клиентами.

   Вторая возможность заключается в определении ключей посредством настройки PGP, которая доступна через Панель Администрирования. В правой стороне этой страницы отображаются все импортированные открытые ключи клиентов. После того как PGP активирован и сконфигурирован, здесь также должен отображатся ваш собственный открытый ключ. В левой части страницы настроек PGP можно производить поиск по ключам. Также, новый открытый ключ может быть загружен в систему из файла.

   Файлы с открытым ключем, которые нужно импортировать в OTRS должны соответствовать файлам ключей GnuPGP. В большинстве случаев, ключ сохраняется в файле как "ASCII armored key". OTRS позволяет работать с этим форматом.

4.3. Безопасность электронной почты с S/MIME

At first glance, encryption with S/MIME seems a little more complicated than with PGP. First, you have to establish a Certification Authority (CA) for the OTRS system. The subsequent steps are very much like those needed with PGP: configure OTRS, install your own certificate, import other public certificates as needed, etc.

В большинстве случаев настройка S/MIME производится за пределами веб-интерфейса OTRS, и должана осуществлятся в оболочке операционной системы с правами пользователя 'otrs'. MIME конфигурация под Linux основана уа SSL (OpenSSL). Поэтому, сначала проверьте установлен ли в вашей системе пакет OpenSSL. OpenSSL-пакет содержит скрипт, который называется CA.pl, для создания сертификата, все наиболее важные шаги, нужно проделать именно с ним. Для упрощения процедуры узнайте, где в файловой системе находится сценарий CA.pl и введите путь к его местоположению в переменную путей оболочки (см. нижеприведенный сценарий).

```
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>
```

Сценарий: Настройка S/MIME.

Сценарий выше показывает, что была создана нова временная директория ~/tmp, в которой хранятся сгенерированные сертификаты.

Для создания сертификата, проделайте следующие операции в командной строке (предположим, что OTRS-администратор должен создать SSL-сертификат с целью тестирования и обучения. В случае, если у вас уже есть SSL-сертификат для шифрования, используйте его и пропустите следующие шаги):

1. Создание собственного Центра Сертификации (Certification Authority) для SSL. Вы должны подтвердить запрос на собственный SSL-сертификат (см. нижеприведенный сценарий).

   ```
   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>
```

Сценарий: Создание Центра Сертификации (Certification Authority) для SSL.

2. Создание запроса сертификата (см. Сценарий ниже).

```
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>
```

Сценарий: Создание запроса сертификаата.

3. Подпись запроса сертификата. Запрос сертификата может быть подписан, и таким образом сертифицирован вашим собственным центром сертификации (CA), что более правдоподобно, чем использование других, внешних CA (см. нижеприведенный сценарий).

```
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>
```

Сценарий: Подпись запроса сертификата.

4. Создание своего собственного сертификата, и все данные собираются с ним, используя запрос сертификата (см. Сценарий ниже).

```
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>
```

Сценарий: Создание нового сертификата.

Теперь эти операции выполнены, установка S/MIME в OTRS должена быть завершена.

Эта часть установки осуществляется в Панели Администрирования, выбрав ссылку "SMIME". В случае, если в OTRS нету поддержки S/MIME, маска указывает администратору удобные ссылки чтобы активировать поддержку S/MIME.

Используя группу "Crypt::SMIME" в SysConfig, также можно включить и настроить поддержку S/MIME.

Здесь можно активировать поддержку S/MIME, определить пути для команд OpenSSL, и директории для сертификатов. Файл ключей должен хранится в директории, указанной здесь. В противном случае OpenSSL не будет их использовать.

Следующим шагом, который следует выполнить - перейти по ссылке Настройка S/MIME в Панели Администрирования. Здесь можно импортировать закрытые ключи OTRS-системы и открытые ключи других партнеров. Введите публичный ключ, который был создан и добавлен в OTRS в начале этого раздела.

Перейдя по ссылке Инструменты администрирования клиентов можно импортировать все открытые S/MIME ключи партнеров по общению.

5. Использование внешних хранилищ данных

5.1. Данные клиентов

OTRS работает с различными атрибутами данных клиентов, такими как имя пользователя, адрес электронной почты, номер телефона и т.д. Эти атрибуты отображаются в обеих фронтэндах, для Агентов и Клиентов. Они также используются для проверки подлинности клиентов.

Данные клиентов, которые используются и отображаются в OTRS очень легко настраиваются. Несмотря на это следующая информация будет всегда необходима для проверки подлинности клиента:

- Вход пользователей
- Адресс электронной почты
- ID Клиента

Use the following SysConfig parameters if you want to display customer information in your agent interface.

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

Script: SysConfig configuration parameters.

5.2. Пользовательский бэк-енд

Вы можете использоваь два типа хранилища информации клиентов: Базу Данных DB и LDAP. Если у вас уже есть другой бэк-енд для хранения пользовательской информации (например SAP), также есть возможность написать модуль для использования этой функции.

5.2.1. База Данных (По умолчанию)

В Примере 11-1 приведена конфигурация базы данных, которая использует данные клиента, хранящиеся в базе данных OTRS.

Пример 4.6. Настройка клиентского хранилища базы данных (DB)

```
# CustomerUser (customer database backend and settings)
$Self->{CustomerUser} = {
    Name => 'Database Datasource',
    Module => 'Kernel::System::CustomerUser::DB',
    Params => {
        # if you want to use an external database, add the required settings
#            DSN => 'DBI:odbc:yourdsn',
#            Type => 'mssql', # only for ODBC connections
#            DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',
#            User => '',
#            Password => '',
            Table => 'customer_user',
            # if your frontend is unicode and the charset of your
            # customer database server is iso-8859-1, use these options.
#           SourceCharset => 'iso-8859-1',
#           DestCharset => 'utf-8',

            # CaseSensitive will control if the SQL statements need LOWER()
            #   function calls to work case insensitively. Setting this to
            #   1 will improve performance dramatically on large databases.
            CaseSensitive => 0,
        },
# 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 => ['title','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
        [ 'UserTitle',      'Title',      'title',      1, 0, 'var', '', 0 ],
        [ 'UserFirstname',  'Firstname',  'first_name', 1, 1, 'var', '', 0 ],
        [ 'UserLastname',   'Lastname',   'last_name',  1, 1, 'var', '', 0 ],
        [ 'UserLogin',      'Username',   'login',      1, 1, 'var', '', 0 ],
        [ 'UserPassword',   'Password',   'pw',         0, 0, 'var', '', 0 ],
        [ 'UserEmail',      'Email',      'email',      1, 1, 'var', '', 0 ],

#        [ 'UserEmail',      'Email', 'email',           1, 1, 'var', '$Env{"CGIHandle"}?
Action=AgentTicketCompose&ResponseID=1&TicketID=$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 => {
        UserTitle => {
            'Mr.' => 'Mr.',
            'Mrs.' => 'Mrs.',
        },
    },
};
```

Если нужно настроить данные клиентов, измените заголовки столбцов или добавьте новые к таблице customer_user базы данных OTRS. В качестве примера используйте нижеприведенный сценарий, в котором показано как добавить новое поле для номера комнаты.

```
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:~#
```

Сценарий: Добавление поля "комната" в таблицу customer_user table.

Теперь добавьте новый столбец в MAP-масив в `Kernel/Config.pm`, как это показано в нижеприведенном сценарии.

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

Сценарий: Добавление поля "комната" в файл Kernel/Config.pm file.

It is also possible to edit all of this customer information via the Customers link in the Agent interface.

5.2.1.1. Клиент с несколькими идентификаторами (Заявки Компании)

Одному клиенту можно назначить больше одного клиентского идентификатора (Customer ID). Это может быть полезно, если клиенту необходимо получить доступ к заявкам других клиентов, например, руководитель хочет посмотреть заявки своих помощников. Если клиент может получить доступ к заявкам другого клиента, то используется особенность OTRS "заявки компании". Заявки компании могут быть доступны перейдя по ссылке "Заявки Компании" в клиентской панели управления.

Для использованя заявок компании, новый столбец с идентификатором (IDs), который должен быть доступен для клиента, должен быть добавлен в таблицу customer_user базы данных OTRS (см. Сценарий ниже).

```
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:~#
```

Сценарий: Добавление поля customer_ids в таблицу customer_user.

Теперь новый столбец должен быть добавлен в MAP-массив в `Kernel/Config.pm`, как это показано в нижеприведенном Сценарии.

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

Сценарий: Добавление поля UserCustomerIDs в Kernel/Config.pm

Теперь, новый столбец для мульти-идентификатора (IDs) клиента можно редактировать с помощью веб-интерфейса Агента, в разделе управления клиентами.

Для того, чтобы убедится что один клиент имеет доступ к заявкам других клиентов нужно добавить идентификаторы (IDs) этих пользователей в новое поле для нескольких идентификаторов клиента. Каждый идентификатор (ID) должен быть отделен точкой с запятой (см. ниже Пример 11-2).

Пример 4.7. Хранение Заявок Компании в базе данных DB

Клиенты А, Б и Ц созданы в вашей системе и А хочет иметь доступ к заявкам Б и Ц используя клиентскую панель. Б и Ц не должны иметь доступа к заявкам других пользователей.

Для реализации этой структуры, измените таблицу customer_user и маппинг (преобразование) в `Kernel/Config.pm` как это показано выше. С помощью Панели Администрирования или используя ссылку Клиенты в веб-интерфейсе Агента загрузите настройки пользователя А. Если настройки отображаются, добавьте значения "Б;Ц;" в поле для CustomerIDs.

5.2.2. LDAP

Если у вас есть LDAP--каталог, в котором хранятся данные о клиентах, его можно использовать в OTRS, в качестве хранилища данных о клиентах, как это показано в Примере 11-3.

Пример 4.8. Настройка LDAP в качестве клиентского бэк-енда

```
# CustomerUser
# (customer ldap backend and settings)
$Self->{CustomerUser} = {
    Name => 'LDAP Data Source',
    Module => 'Kernel::System::CustomerUser::LDAP',
    Params => {
        # ldap host
        Host => 'bay.csuhayward.edu',
        # ldap base dn
        BaseDN => 'ou=seas,o=csuh',
        # search scope (one|sub)
        SSCOPE => 'sub',
        # The following is valid but would only be necessary if the
        # anonymous user does NOT have permission to read from the LDAP tree
        UserDN => '',
        UserPw => '',
        # in case you want to add always one filter to each ldap query, use
        # this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter => '(objectclass=user)'
        AlwaysFilter => '',
            # if both your frontend and your LDAP are unicode, use this:
            SourceCharset => 'utf-8',
            DestCharset   => 'utf-8',
            # if your frontend is unicode and the charset of your
            # ldap server is iso-8859-1, use these options.
            # SourceCharset => 'iso-8859-1',
            # DestCharset => 'utf-8',
            # Net::LDAP new params (if needed - for more info see perldoc Net::LDAP)
            Params => {
                port => 389,
                timeout => 120,
                async => 0,
                version => 3,
            },
    },
    # customer unique id
    CustomerKey => 'uid',
    # customer #
    CustomerID => 'mail',
    CustomerUserListFields => ['cn', 'mail'],
    CustomerUserSearchFields => ['uid', 'cn', 'mail'],
    CustomerUserSearchPrefix => '',
    CustomerUserSearchSuffix => '*',
    CustomerUserSearchListLimit => 250,
    CustomerUserPostMasterSearchFields => ['mail'],
    CustomerUserNameFields => ['givenname', 'sn'],
    # show not own tickets in customer panel, CompanyTickets
    CustomerUserExcludePrimaryCustomerID => 0,
    # add an ldap filter for valid users (expert setting)
#    CustomerUserValidFilter => '(!(description=locked))',
    # administrator can't change customer preferences
    AdminSetPreferences => 0,
#    # cache time to live in sec. - cache any database queries
#    CacheTTL => 0,
    Map => [
        # note: Login, Email and CustomerID are mandatory!
        # var, frontend, storage, shown (1=always,2=lite), required, storage-type, http-link,
 readonly
        [ 'UserTitle',      'Title',      'title',           1, 0, 'var', '', 0 ],
        [ 'UserFirstname',  'Firstname',  'givenname',       1, 1, 'var', '', 0 ],
        [ 'UserLastname',   'Lastname',   'sn',              1, 1, 'var', '', 0 ],
        [ 'UserLogin',      'Username',   'uid',             1, 1, 'var', '', 0 ],
        [ 'UserEmail',      'Email',      'mail',            1, 1, 'var', '', 0 ],
        [ 'UserCustomerID', 'CustomerID', 'mail',            0, 1, 'var', '', 0 ],
#        [ 'UserCustomerIDs', 'CustomerIDs', 'second_customer_ids', 1, 0, 'var', '', 0 ],
        [ 'UserPhone',      'Phone',      'telephonenumber', 1, 0, 'var', '', 0 ],
        [ 'UserAddress',    'Address',    'postaladdress',   1, 0, 'var', '', 0 ],
        [ 'UserComment',    'Comment',    'description',     1, 0, 'var', '', 0 ],
    ],
};
```

Если дополнительные атрибуты клиента хранятся в LDAP-каталоге, например, имя руководителя, номер мобильного телефона, или отдела, и если эту информацию нужно отобразить в OTRS, просто расширьте MAP-массив в файле `Kernel/Config.pm` записями для этих атрибутов, как показано в нижеприведенном сценарии.

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

Сценарий: Добавление нового поля в файл Kernel/Config.pm.

5.2.2.1. Клиент с несколькими идентификаторами (Заявки Компании)

При использовании LDAP-бэкэнда клиенту можно присвоить больше одного клиентского айди (Customer ID). Для использования заявок компании, в LDAP-директорию нужно добавить новое поле, которое содержит доступные агенту идентификаторы (IDs).

Если в LDAP-каталоге было созадно новое поле, его также нужно добавить в MAP-массив в `Kernel/Config.pm`, как это показано в нижеприведенном сценарии.

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

Сценарий: Подстановка новых полей в Kernel/Config.pm file.

Клиентские идентификаторы (IDs) можно редактировать напрямую в LDAP-директории. OTRS может только считать информацию из LDAP, но не записывать.

Чтобы убедтися что клиент имеет доступ к заявкам других клиентов, добавьте идентификаторы (IDs) клиентов, к заявкам которых нужен доступ, к новому полю вашей LDAP-директории. Каждый идентификатор ID должен быть отделен точнкой с запятой (см. ниже Рисунок 11-4).

Пример 4.9. Использование заявок Компании с LDAP-бэкэндом

Клиенты А, Б и Ц созданны в вашей системе и А хочет иметь доступ к заявкам Б и Ц через панель клиента. Б и Ц не должны иметь доступа к другим пользователям.

Для реализации этой установки измените LDAP-директорию и маппинг (mapping) в `Kernel/Config.pm`, как это показано выше. Затем добавьте в вашей LDAP-директории значения "Б;Ц;" в поле для CustomerIDs, для клиента "А".

5.2.3. Использование больше чем одного пользовательского хранилища данных с OTRS

Если в OTRS нужно использовать больше одного источника данных о клиентах (например LDAP и базу данных), конфигурационный параметр CustomerUser должен быть расширен числом, например "CustomerUser1", "CustomerUser2" (см. ниже Пример 11-5).

Пример 4.10. Использование больше чем одного пользовательского хранилища данных с OTRS

В следующем примере показано применение конфигурации как для LDAP так и для базы данных клиентского бэкэнда с OTRS.

```
# 1. Customer user backend: DB
# (customer database backend and settings)
$Self->{CustomerUser1} = {
    Name => 'Customer Database',
    Module => 'Kernel::System::CustomerUser::DB',
    Params => {
        # if you want to use an external database, add the
        # required settings
#        DSN => 'DBI:odbc:yourdsn',
#        Type => 'mssql', # only for ODBC connections
#        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 => ['title','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
        [ 'UserTitle',      'Title',      'title',        1, 0, 'var', '', 0 ],
        [ 'UserFirstname',  'Firstname',  'first_name',   1, 1, 'var', '', 0 ],
        [ 'UserLastname',   'Lastname',   'last_name',    1, 1, 'var', '', 0 ],
        [ 'UserLogin',      'Username',   'login',        1, 1, 'var', '', 0 ],
        [ 'UserPassword',   'Password',   'pw',           0, 0, 'var', '', 0 ],
        [ 'UserEmail',      'Email',      'email',        1, 1, 'var', '', 0 ],
        [ 'UserCustomerID', 'CustomerID', 'customer_id',  0, 1, 'var', '', 0 ],
        [ 'UserPhone',      'Phone',      'phone',        1, 0, 'var', '', 0 ],
        [ 'UserFax',        'Fax',        'fax',          1, 0, 'var', '', 0 ],
        [ 'UserMobile',     'Mobile',     'mobile',       1, 0, 'var', '', 0 ],
        [ 'UserStreet',     'Street',     'street',       1, 0, 'var', '', 0 ],
        [ 'UserZip',        'Zip',        'zip',          1, 0, 'var', '', 0 ],
        [ 'UserCity',       'City',       'city',         1, 0, 'var', '', 0 ],
        [ 'UserCountry',    'Country',    'country',      1, 0, 'var', '', 0 ],
        [ 'UserComment',    'Comment',    'comments',     1, 0, 'var', '', 0 ],
        [ 'ValidID',        'Valid',      'valid_id',     0, 1, 'int', '', 0 ],
    ],
    # default selections
    Selections => {
        UserTitle => {
            'Mr.' => 'Mr.',
            'Mrs.' => 'Mrs.',
        },
    },
};

# 2. Customer user backend: LDAP
# (customer ldap backend and settings)
$Self->{CustomerUser2} = {
    Name => 'LDAP Datasource',
    Module => 'Kernel::System::CustomerUser::LDAP',
    Params => {
        # ldap host
        Host => 'bay.csuhayward.edu',
        # ldap base dn
        BaseDN => 'ou=seas,o=csuh',
        # search scope (one|sub)
        SSCOPE => 'sub',
#        # The following is valid but would only be necessary if the
#        # anonymous user does NOT have permission to read from the LDAP tree
        UserDN => '',
        UserPw => '',
        # in case you want to add always one filter to each ldap query, use
        # this option. e. g. AlwaysFilter => '(mail=*)' or AlwaysFilter => '(objectclass=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
```

```
        [ 'UserTitle',      'Title',      'title',           1, 0, 'var', '', 0 ],
        [ 'UserFirstname',  'Firstname',  'givenname',       1, 1, 'var', '', 0 ],
        [ 'UserLastname',   'Lastname',   'sn',              1, 1, 'var', '', 0 ],
        [ 'UserLogin',      'Username',   'uid',             1, 1, 'var', '', 0 ],
        [ 'UserEmail',      'Email',      'mail',            1, 1, 'var', '', 0 ],
        [ 'UserCustomerID', 'CustomerID', 'mail',            0, 1, 'var', '', 0 ],
#        [ 'UserCustomerIDs', 'CustomerIDs', 'second_customer_ids', 1, 0, 'var', '', 0 ],
        [ 'UserPhone',      'Phone',      'telephonenumber', 1, 0, 'var', '', 0 ],
        [ 'UserAddress',    'Address',    'postaladdress',   1, 0, 'var', '', 0 ],
        [ 'UserComment',    'Comment',    'description',     1, 0, 'var', '', 0 ],
    ],
};
```

Есть возможность интегрировать до 10 различных пользовательских бэк-эндов. Используйте интерфейс управления клиентами в OTRS, чтобы просматривать и редактировать данные о них (при условии наличия прав для записи).

5.2.4. Storing CustomerUser data in dynamic fields

Sometimes it can be useful to also store CustomerUser data directly in dynamic fields of a ticket, for example to create special statistics on this data.

The dynamic field values are set when a ticket is created or when the customer of a ticket is changed. The values of the dynamic fields are taken from the customer data. This works for all backends, but is especially useful for LDAP-backends.

To activate this optional feature of OTRS, please activate the settings "Ticket::EventModulePost###930-DynamicFieldFromCustomerUser" and "DynamicFieldFromCustomerUser::Mapping". The latter setting contains the configuration of which CustomerUser field entry should be stored in which ticket dynamic field. The fields must be present in the system and should be enabled for AgentTicketFreeText, so that they can be set manually. They mustn't be enabled for AgentTicketPhone, AgentTicketEmail and AgentTicketCustomer. If they were, they would have precedence over the automatically set values.

5.3. Хранилища (бэк-енды) для аутентификации Агентов и Клиентов

OTRS предлагает опцию для проверки подлинности агентов и клиентов с использованием различных хранилищ данных (бэкендов).

5.3.1. Хранилища данных (бэк-енды) для аутентификации Агентов

5.3.1.1. База Данных (DB, по умаолчанию)

В качестве бэк-энда для аутентификации агентов в OTRS, по умолчанию, используется база данных. Чтобы добавлять агентов, редактировать данные о них, перейдите на страницу Панель Администрирования и нажмите ссылку Интерфейс для управления агентами (см. ниже Пример 11.6).

Пример 4.11. Проверка подлинности агентов путем использования Базы Данных (DB) в качестве хранилища информации.

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

5.3.1.2. LDAP

Если данные всех ваших агентов хранятся в LDAP-директории, то для аутентификации пользователей в OTRS можно использовать LDAP-модуль (см. ниже Пример 11-7). Этот модуль имеет права только на чтение дерева LDAP-каталогов, что означает что нету возможности редактировать данные пользователей используя ссылку веб-интерфейс для управления пользователями .

Пример 4.12. Проверка подлинности агентов при использовании LDAP в качестве хранилища данных

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

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

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

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

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

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

The configuration parameters shown in the script below 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 synchronization is done when the agent authenticates the first time. Although the data can be syncronized 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.

```
# defines AuthSyncBackend (AuthSyncModule) for AuthModule
# if this key exists and is empty, there won't be a sync.
# example values: AuthSyncBackend, AuthSyncBackend2
$Self->{'AuthModule::UseSyncBackend'} = 'AuthSyncBackend';

# 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',
];
```

Сценарий: Синхронизация пользовательских данных из LDAP-директориив базу данных OTRS.

5.3.1.3. HTTPBasicAuth-аутентификация для Агентов

Если вы хотите реализовать решение "single sign on" для всех агентов, вы можете использовать базовую аунентификацию (для всех систем) и HTTPBasicAuth-модуль для OTRS (см. ниже Пример 11-8).

Пример 4.13. Аутентификация Агентов с помощью HTTPBasic

```
# This is an example configuration for an apache ($ENV{REMOTE_USER})
# auth. backend. Use it if you want to have a singe 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-authorised-for-otrs.html';
$Self->{LogoutURL} = 'http://host.example.com/thanks-for-using-otrs.html';
```

5.3.1.4. Radius

Параметры конфигурации приведенные в Примере 11-9 могут быть использованы для аутентификации агентов с использованием Radius-сервера.

Пример 4.14. Аутентификация (проверка подлинности) агентов с использованием Radius-сервера в качестве хранилища информации

```
# 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';
```

5.3.2. Хранилища информации для аутентификации (проверки подлинности) клиентов

5.3.2.1. База Данных (По умолчанию)

Для аутентификации клиентов в OTRS, по умолчанию, используется база данных. Используя базу данных в качестве хранилища, все данные клиентов можно редактировать через веб-интерфейс OTRS (см. ниже Пример 11-10).

Пример 4.15. Аутентификация Клиента в Базе Данных

```
# 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";
```

5.3.2.2. LDAP

Если у вас есть LDAP-каталог со всеми данными о клиентах, можно использовать модуль LDAP для аутентификации клиентов в OTRS (см. Пример 11-11 ниже). Поскольку этот модуль имеет права только для чтения для данных из LDAP-бэкенда, то нету возможности изменить данные клиента через веб.

Пример 4.16. Аутентификация пользователей с помощью LDAP-бэкэнда

```
# 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,
};
```

5.3.2.3. HTTPBasicAuth аутентификация для клиентов

Если для пользователей нужно внедрить "single sign on"-решение, можно использовать базовую, HTTPBasic аутентификацию (для всех систем) и использовать модуль HTTPBasicAuth с OTRS (больше не нужно логинится в OTRS). См. ниже Пример 11-12.

Пример 4.17. Аутентификация клиентов с помощью HTTPBasic

```
# This is an example configuration for an apache ($ENV{REMOTE_USER})
# auth. backend. Use it if you want to have a singe 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';
```

5.3.2.4. Radius

Настройки приведенные в примере 11-13 могут быть использованы для аутентификации ваших клиентов с помощью Radius-сервера.

Пример 4.18. Аутентификация клиентов с использованием Radius

```
# 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';
```

5.4. Настройка самостоятельной регистрации клиента

Есть возможность настроить самостоятельную регистрацию новых клиентов, доступную через панель customer.pl. Можно добавить новые дополнительные или обязательные поля, такие как номер комнаты, адрес или состояние.

В следующем примере показано каким образом можно указать обязательные для заполнения поля в базе данных клиентов, в данном случае, для хранения номера клиента.

5.4.1. Настройка веб-интерфейса

Для отображения нового поля для номера комнаты в веб-интерфейсе пользователя (customer.pl), нужно внести изменения в .dtl-файл, который отвечает за шаблон этого интерфейса. Отредактируйте файл Kernel/Output/HTML/Standard/CustomerLogin.dtl добавив новые поля после строки 80 (см. нижеприведенный сценарий).

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

Сценарий: Отображение новых полей в веб-интерфейсе.

5.4.2. Отображения клиентов

На следующем этапе, нужно внести изменения в маппинг (mapping), расширив его новым элементом - номером комнаты. Чтобы убедится что изменения не потеряются после обновления, установите настройки "CustomerUser" из файла Kernel/Config/Defaults.pm в файл Kernel/Config.pm. Теперь измените MAP-массив, добавив новое поле - номер комнаты, как это показано в нижеприведенном сценарии.

```
# CustomerUser
# (customer database backend and settings)
$Self->{CustomerUser} = {
    Name => 'Database Backend',
    Module => 'Kernel::System::CustomerUser::DB',
    Params => {
        # if you want to use an external database, add the
        # required settings
#        DSN => 'DBI:odbc:yourdsn',
#        Type => 'mssql', # only for ODBC connections
#        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 => ['title', '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
        [ 'UserTitle',      'Title',      'title',       1, 0, 'var', '', 0 ],
        [ 'UserFirstname',  'Firstname',  'first_name',  1, 1, 'var', '', 0 ],
        [ 'UserLastname',   'Lastname',   'last_name',   1, 1, 'var', '', 0 ],
        [ 'UserLogin',      'Username',   'login',       1, 1, 'var', '', 0 ],
        [ 'UserPassword',   'Password',   'pw',          0, 0, 'var', '', 0 ],
        [ 'UserEmail',      'Email',      'email',       1, 1, 'var', '', 0 ],
        [ 'UserCustomerID', 'CustomerID', 'customer_id', 0, 1, 'var', '', 0 ],
        [ 'UserPhone',      'Phone',      'phone',       1, 0, 'var', '', 0 ],
        [ 'UserFax',        'Fax',        'fax',         1, 0, 'var', '', 0 ],
        [ 'UserMobile',     'Mobile',     'mobile',      1, 0, 'var', '', 0 ],
        [ 'UserRoom',       'Room',       'room',        1, 0, 'var', '', 0 ],
        [ 'UserStreet',     'Street',     'street',      1, 0, 'var', '', 0 ],
        [ 'UserZip',        'Zip',        'zip',         1, 0, 'var', '', 0 ],
        [ 'UserCity',       'City',       'city',        1, 0, 'var', '', 0 ],
        [ 'UserCountry',    'Country',    'country',     1, 0, 'var', '', 0 ],
        [ 'UserComment',    'Comment',    'comments',    1, 0, 'var', '', 0 ],
        [ 'ValidID',        'Valid',      'valid_id',    0, 1, 'int', '', 0 ],
    ],
    # default selections
    Selections => {
        UserTitle => {
            'Mr.' => 'Mr.',
            'Mrs.' => 'Mrs.',
        },
    },
};
```

Сценарий: Внесение изменений в map-массив.

5.4.3. Настройка таблицы customer_user в Базе Данных OTRS DB

Последним шагом является добавление нового столбца с номером комнаты в таблицу customer_user базы данных OTRS (см. нижеприведенный сценарий). В этой колонке будет хранится информация для номера комнаты.

```

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:~#
```

Сценарий: Добавление новой колонки в таблицу customer_user.

Now the new field for the room should be displayed in the Customer Information panel if filled, and in the Customer User administration screens. Also, new customers should have to insert their room number if they register a new account. If you use OTRS on Microsoft IIS, you should restart the web server to activate the changes made in `Config.pm`.

6. Настройки Заявки

6.1. Состояния заявок

6.1.1. Предопределенные состояния

OTRS позволяет изменять предопределенные состояния заявок и их типы, а также добавлять новые. Для состояния важны два атрибута: имя (state-name) и тип (state-type).

Предустановленные состояния в OTRS: "закрыто успешно", "закрыто неудачно", "обьеденено", "новая", "открытая", "в ожидании с автозакрытием+", "в ожидании с автозакрытием-", "в ожидании с напоминанием", "удаленная".

6.1.1.1. Новая

Заявки находятся в этом состоянии, когда они создаются на основе входящих сообщений электронной почты.

6.1.1.2. Открытая

Это состояние по умолчанию для заявок, которые присвоены очередям или агентам.

6.1.1.3. Ожидание с напоминанием

После того как время ожидания истекло, владелец заявки будет получать напоминание на электронную почту. Если заявка не закрыта, то напоминание о заявке будет отправлено всем агентам в очереди. Напоминание о заявках будет отправлено только в рабочее время и будет повторятся каждые 24-часа, пока агент не изменит состояние заявки. Время, которое заявка проведет с таким статусом будет добавлено к времени эскалации.

6.1.1.4. Ожидание авто-закрытие -

Если время ожидания вышло, заявки с этим статусом будут установлены в "Закрытые неуспешно". Время, проведенное заявкой в этом статусе будет добавлено к времени эскалации.

6.1.1.5. Ожидание авто-закрытие+

Если вышло время ожидания, заявки с этим статусом будут установлены в "Закрыто Успешно". Время, проведенное заявкой в этом статусе будет добавлено к времени эскалации.

6.1.1.6. Обьедененные

Это состояние для заявок, которые были объеденены с другими заявками.

6.1.1.7. Закрыта Успешно

Это конечное состояние для заявок, которые были решены успешно. В зависимости от конфигурации, у вас будет или не будет возможности заново открыть ранее закрытые заявки.

6.1.1.8. Закрыта Неудачно

Это конечное состояние для заявок которые НЕ были успешно решены. В зависимости от конфигурации, у вас будет или не будет возможности заново открыть ранее закрытые заявки.

6.1.2. Настраиваемы состояния

Каждое состояние имеет название (state-name) и тип (state-type). Чтобы создать новое состояние перейдите по ссылке Состояния на Панели Администрирования и нажмите кнопку "Добавить состояние". Можно свободно выбирать имя нового состояния. Типы состояний не могут изменятся посредством веб-интерфейса. Если нужно добавить новые типы или изменить существующие, - все изменения придется делать напрямую в базе данных. Предустановленные типы состояний не могут быть изменены, поскольку это может привести к непредсказуемым результатам. Например, расчет эскалации и фунция разблокирования основаны на конкретных типах состояний.

The name of an already existing state can be changed, or new states added through this screen. If the state "new" has been changed via the web interface, this change also has to be configured via the config file `Kernel/Config.pm` or via the SysConfig interface. The settings specified in the script below 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';
    [...]
```

Сценарий: Изменение параметров настройки в Kernel/Config.pm.

Если нужно добавить новый тип состояния, то это можно сделать с помощью клиентской программы управления базами данных, изменив таблицу ticket_state_type базы данных OTRS, как это показано в нижеприведенном сценарии

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

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

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

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

mysql> quit
Bye
linux:~#
```

Script: Изменение базы данных OTRS.

На данный момент можно использовать новый тип состояния, который вы только что создали. Как только состояние будет связано с этим новым типом состояния, то чтобы убедится что новое состояние используется и работает коректно нужно также изменить настройки OTRS. Используя SysConfig внесите изменения в следующие опции:

Ticket -> Frontend::Agent::Ticket::ViewPhoneNew > AgentTicketPhone###StateDefault - определить следующее состояние по умолчанию для новых заявок созданных на основе телефонного звонка.

Ticket -> Frontend::Agent::Ticket::ViewPhoneNew > AgentTicketPhone###StateType - для определения последующих доступных состояний для новых заявок, созданных на основе телефонного звонка.

Ticket -> Frontend::Agent::Ticket::ViewEmailNew > AgentTicketEmail###StateDefault - установка последующих следующих состояний для заявок созданных на базе сообщений электронной почты.

Ticket -> Frontend::Agent::Ticket::ViewEmailNew > AgentTicketEmail###StateType - для определения последующих доступных состояний для новых заявок созданных на базе сообщений электронной почты.

Ticket -> Frontend::Agent::Ticket::ViewPhoneOutbound > AgentTicketPhoneOutbound###State - для определения последующих доступных состояний для новых заявок созданных на базе новых телефонных статей.

Ticket -> Frontend::Agent::Ticket::ViewPhoneOutbound > AgentTicketPhoneOutbound###StateType - для определения последующих доступных состояний для новых заявок созданных на базе новых телефонных статей.

Ticket:Frontend::Agent::Ticket::ViewMove:Ticket::DefaultNextMoveStateType - для определения последующих доступных состояний для перемещенных заявок.

Ticket -> Frontend::Agent::Ticket::ViewBounce > StateDefault - для определения последующих доступных состояний для "подпрыгивающих" заявок.

Ticket -> Frontend::Agent::Ticket::ViewBounce > StateType - для определения последующих доступных состояний для экранов с отказами.

Ticket -> Frontend::Agent::Ticket::ViewBulk > StateDefault - для определения прдеопределенных последующих состояний для груповых действий.

Ticket -> Frontend::Agent::Ticket::ViewBulk > StateType - для определения прдеопределенных последующих состояний для экрана груповых действий

Ticket -> Frontend::Agent::Ticket::ViewClose > StateDefault - для определения прдеопределенных последующих состояний после закрытия заявки

Ticket -> Frontend::Agent::Ticket::ViewClose > StateType - для определения прдеопределенных последующих состояний для страницы закрытия.

Ticket -> Frontend::Agent::Ticket::ViewCompose > StateDefault - для определения прдеопределенных последующих состояний для Compose-страницы

Ticket -> Frontend::Agent::Ticket::ViewCompose > StateType - для определения прдеопределенных последующих состояний для Compose-страницы.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateDefault - для определения прдеопределенных последующих состояний после перенаправления заявки.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateType - для определения прдеопределенных последующих состояний для страници перенаправления.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateDefault - для определения прдеопределенных последующих состояний для страницы free-text.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateType - для определения прдеопределенных последующих состояний для free text-страницы.

Ticket -> Core::PostMaster > PostmasterDefaultState - для определения прдеопределенных последующих состояний для заявок, созданых с сообщений электронной почты.

Ticket -> Core::PostMaster > PostmasterFollowUpState - для определения прдеопределенных последующих состояний для заявок после последующих, которые должны быть сохранены.

Ticket -> Core::PostMaster > PostmasterFollowUpStateClosed - для определения состояния заявок, после того как придет новый ответ на уже закрытую заявку.

Ticket -> Core::Ticket > ViewableStateType - для определения состояний, которые будут отображатся в различных местах системы, напримерв в Queueview.

Ticket -> Core::Ticket > UnlockStateType - для определения типов состояний для разблокированых заявок.

Ticket -> Core::Ticket > PendingReminderStateType - для определения типов состояний для заявок с напоминанием.

Ticket -> Core::Ticket > PendingAutoStateType - для определения типов состояний для ожидающих заявок.

Ticket -> Core::Ticket > StateAfterPending - для определения состояния заявки установленой в Таймере Авто-Ожидания, если время для настроенного состояния истекло.

6.2. Приоритеты заявок

OTRS поставляется с пятью предустановлеными уровнями приоритетов, которые можно изменить перейдя по ссылке "Приоритеты" на Панели Администрирования. При создании настраиваемого списка приоритетов, пожалуйста помните, что они сортируются в алфавитном порядке. Также OTRS сортирует заявки в QueueView по их внутреннему номеру (ID).

Примечание

Как и в случае с другими сущностями OTRS, приоритеты не могут быть удалены, а только деактивированы путем установки параметра Действительный в значение *не действительный* или *не действительный-временно*.

Важно

Если был создан новый приоритет, или был изменен уже существующий, то можно также произвести изменения некоторых параметров в SysConfig:

- Ticket:Core::Postmaster::PostmasterDefaultPriority - определяет предустановленный приоритет для всех входящих сообщений электронной почты.

- Ticket:Frontend::Agent:Ticket::ViewPhoneNew:Priority - определяет предустановленный приоритет для новых Заявок созданных на основе Телефонных Звонков.

- Ticket:Frontend::Agent:Ticket::ViewEmailNew:Priority - определяет предустановленый приоритет на странице с новыми Email-Заявками для агентов.

- Ticket:Frontend::Customer:Ticket::ViewNew:PriorityDefault - определяет предустановленные приоритеты для страцицы Новые Заявки в пользовательском веб-интерфейсе.

6.3. Ответственность за Заявку & Наблюдение за Заявкой

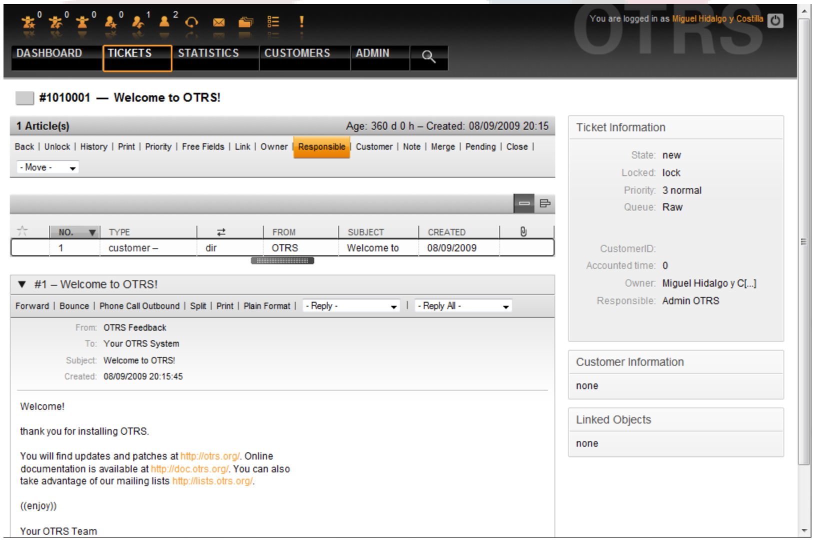
From OTRS 2.1 on, it is possible to assign a person as being responsible for a ticket, in addition to its owner. Moreover, all activities connected with the ticket can be watched by someone other than the ticket owner. These two functionalities are implemented with the TicketResponsible and TicketWatcher features, and facilitate the assignment of tasks and working within hierarchical team structures.

6.3.1. Ответственность за Заявку

Функция ответственности за заявку способствует ее полной обработки еагентом, который не является владельцем заявки. Таким образом агент, который заблокировал заявку может передать ее другому агенту, который не является владельцем заявки, для того, чтобы второй просто дал ответ на вопрос. После того как запрос был рассмотрен, первый агент может снять ответственность за заявку с второго агента.

С помощью параметров конфигурации Ticket::Responsible, можно активировать функцию ответственности за заявку. Это приведет к появлению на экране трех дополнительных иконок/значков.

Ответственность за заявку может быть назначена после открытия ее содержимого, нажав ссылку "Ответственность", соответсвующего меню в шаблоне просмотра подробной информации о заявке агентского веб-интерфейса (см. нижеприведенный сценарий).


You are logged in as Miguel Hidalgo y Costilla
DASHBOARD
TICKETS
STATISTICS
CUSTOMERS
ADMIN
Back | Unlock | History | Print | Priority | Free Fields | Link | Owner | Responsible | Customer | Note | Merge | Pending | Close |
- Move -
Forward | Bounce | Phone Call Outbound | Split | Print | Plain Format |
- Reply -
- Reply All -

#1010001 — Welcome to OTRS!

1 Article(s) Age: 360 d 0 h – Created: 08/09/2009 20:15

NO.	TYPE	⇄	FROM	SUBJECT	CREATED
1	customer –	dir	OTRS	Welcome to	08/09/2009

#1 – Welcome to OTRS!

From: OTRS Feedback
To: Your OTRS System
Subject: Welcome to OTRS!
Created: 08/09/2009 20:15:45

Welcome!

thank you for installing OTRS.

You will find updates and patches at http://otrs.org/. Online documentation is available at http://doc.otrs.org/. You can also take advantage of our mailing lists http://lists.otrs.org/.

((enjoy))

Your OTRS Team

Ticket Information

State: new
Locked: lock
Priority: 3 normal
Queue: Raw
CustomerID:
Accounted time: 0
Owner: Miguel Hidalgo y C[...]
Responsible: Admin OTRS

Customer Information

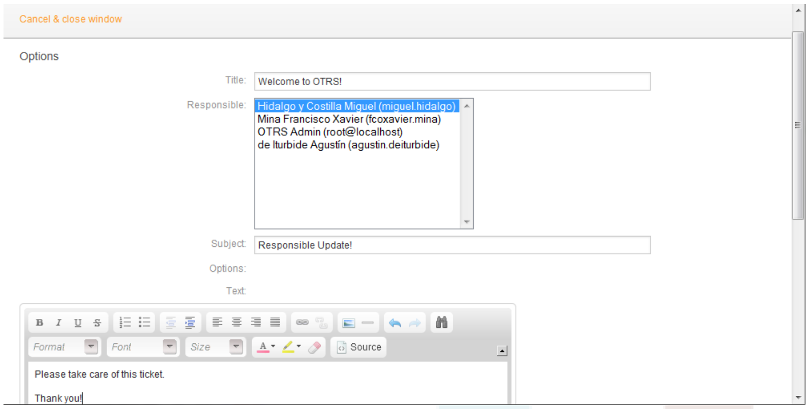
none

Linked Objects

none

Рисунок: Изменение Ответственного за заявку в шаблоне просмотра подробной инф. о заявке.

После нажатия на кнопку "Ответственность", откроется всплывающее окно для изменения ответственности этой заявки (см. нижеприведенный Рисунок). Этот шаблон/диалог также может быть использован для отправки сообщения новому агенту, который будет нести ответственность за эту заявку.



Cancel & close window

Options

Title: Welcome to OTRS!

Responsible:
Hidalgo y Costilla Miguel (miguel.hidalgo)
Mina Francisco Xavier (fcoxavier.mina)
OTRS Admin (root@localhost)
de Iturbide Agustín (agustin.deiturbide)

Subject: Responsible Update!

Options:

Text:

Please take care of this ticket.

Thank you!

Рисунок: Всплывающий диалог для изменения ответственного за заявку.

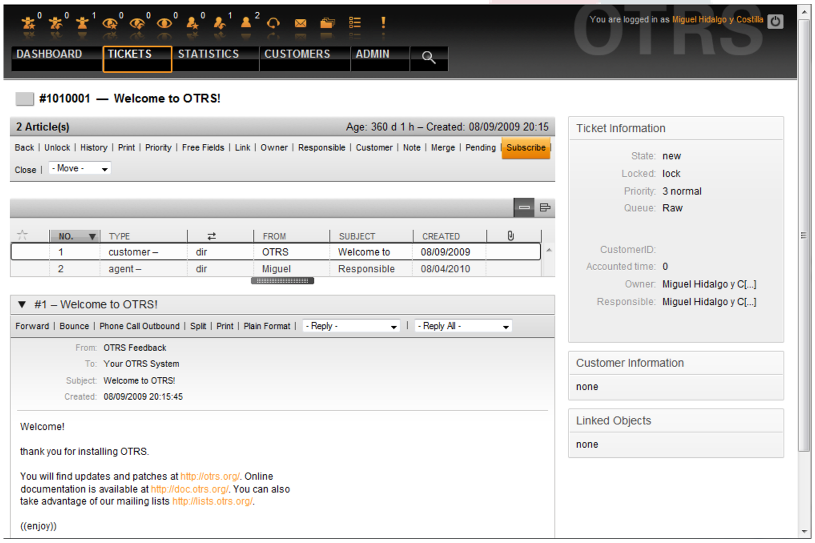
Если активирована функция ответственности за заявку, то в шаблоне Ответственность, агентского веб-интерфейса OTRS можно просмотреть список всех заявок, за которые агент несет ответственность.

6.3.2. Просмотр Заявок

Начиная с OTRS 2.1 и выше с помощью функции TicketWatcher, выбранные агенты, такие как, например, руководители могут просматривать определенные заявки без их обработки.

Функция TicketWatcher может быть активирована с помощью параметра конфигурации Ticket::Watcher, после чего в панели инструментов появлятся новые ссылки. Используя Ticket::WatcherGroup можно определить одну или несколько групп пользователей с правами просмотра заявок.

Для того чтобы смотреть заявку, перейдите к шаблону просмотра подробной информации о заявке и нажмите ссылку "Подписатся" в меню заявки (см. нижеприведенный Рисунок).



You are logged in as Miguel Hidalgo y Costilla

#1010001 — Welcome to OTRS!

2 Article(s) Age: 360 d 1 h – Created: 08/09/2009 20:15

Back | Unlock | History | Print | Priority | Free Fields | Link | Owner | Responsible | Customer | Note | Merge | Pending | Subscribe |

Close | - Move -

NO.	TYPE	⇄	FROM	SUBJECT	CREATED
1	customer –	dir	OTRS	Welcome to	08/09/2009
2	agent –	dir	Miguel	Responsible	08/04/2010

▼ #1 – Welcome to OTRS!

Forward | Bounce | Phone Call Outbound | Split | Print | Plain Format | - Reply - | - Reply All -

From: OTRS Feedback
To: Your OTRS System
Subject: Welcome to OTRS!
Created: 08/09/2009 20:15:45

Welcome!

thank you for installing OTRS.

You will find updates and patches at http://otrs.org/. Online documentation is available at http://doc.otrs.org/. You can also take advantage of our mailing lists http://lists.otrs.org/.

((enjoy))

Ticket Information

State: new
Locked: lock
Priority: 3 normal
Queue: Raw

CustomerID:
Accounted time: 0
Owner: Miguel Hidalgo y C[...]
Responsible: Miguel Hidalgo y C[...]

Customer Information

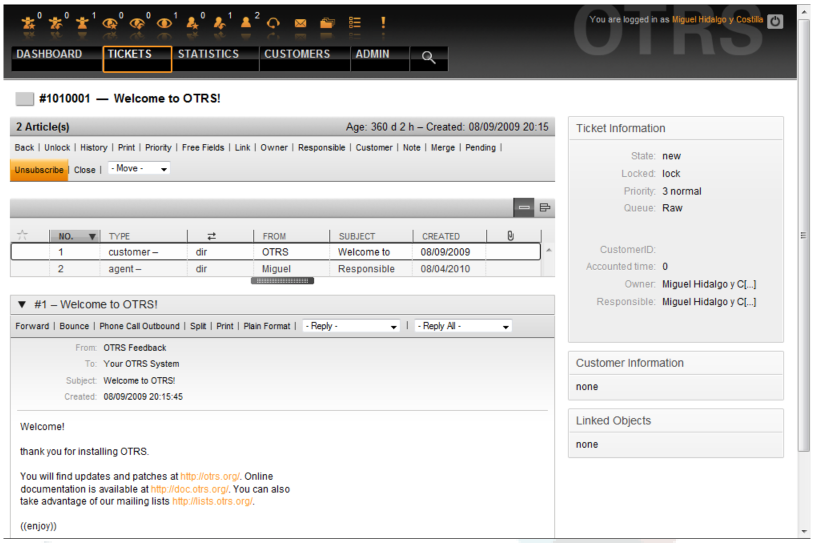
none

Linked Objects

none

Figure: Подписка на просмотр подробной информации о заявке.

Если вы больше не хотите просматривать определенную заявку, перейдите к шаблону просмотра подробной информации заявки и нажмите кнопку "Отменить подписку" в меню заявки (см. нижеприведенный рисунок).


DASHBOARD
TICKETS
STATISTICS
CUSTOMERS
ADMIN
You are logged in as Miguel Hidalgo y Costilla
Back | Unlock | History | Print | Priority | Free Fields | Link | Owner | Responsible | Customer | Note | Merge | Pending |
Unsubscribe | Close | - Move -
Forward | Bounce | Phone Call Outbound | Split | Print | Plain Format | - Reply - | - Reply All -

#1010001 — Welcome to OTRS!

2 Article(s) Age: 360 d 2 h – Created: 08/09/2009 20:15

NO.	TYPE	⇄	FROM	SUBJECT	CREATED
1	customer –	dir	OTRS	Welcome to	08/09/2009
2	agent –	dir	Miguel	Responsible	08/04/2010

▼ #1 – Welcome to OTRS!

From: OTRS Feedback
To: Your OTRS System
Subject: Welcome to OTRS!
Created: 08/09/2009 20:15:45

Welcome!

thank you for installing OTRS.

You will find updates and patches at http://otrs.org/. Online documentation is available at http://doc.otrs.org/. You can also take advantage of our mailing lists http://lists.otrs.org/.

((enjoy))

Ticket Information

State: new
Locked: lock
Priority: 3 normal
Queue: Raw
CustomerID:
Accounted time: 0
Owner: Miguel Hidalgo y C[...]
Responsible: Miguel Hidalgo y C[...]

Customer Information

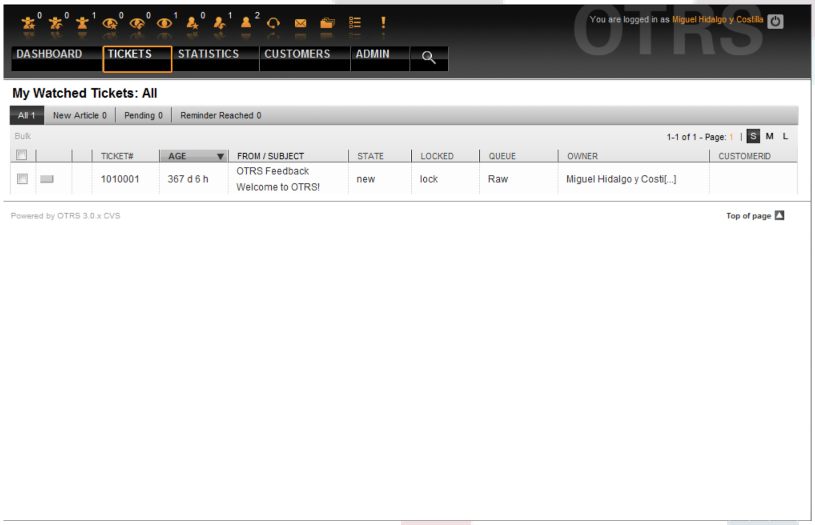
none

Linked Objects

none

Рисунок: Отказ от подписи на заявку в шаблоне просмотра подробной информации о заявке.

Как только активирована функция "просмотр заявок", то список всех просматриваемых заявок будет доступен в шаблоне Watched View (см. нижеприведенный Рисунок).



My Watched Tickets: All

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

Powered by OTRS 3.0.x CVS

Рисунок: Шаблон для просматриваемых заявок.

7. Функции связанные с временем

7.1. Настройка периода рабочего времени, празников и часовых поясов

Некоторые функции в OTRS, например, как эскалация и автоматическая разблокировка заявок зависят от правильной настройки параметра "рабочее время", часовых поясов и праздников. Все эти настройки можно определить перейдя по ссылке интерфейс SysConfig в Framework > Core::Time. Перейдя по ссылке Framework > Core::Time::Calendar1, которая находится подFramework > Core::Time::Calendar9 можно также указать различные, заранее установленные "наборы рабочего времени", праздников и часовых поясов. Перейдя по ссылке настройки очереди можно определить Календари. Для этого также можно использовать SLA-уровни. Например можно создать различные SLA-схемы: 5 x 8 рабочих часов для стандартного SLA, но и вместе с тем создать другой календарь: 7 x 24 - "Золотой SLA"; также можно иметь отдельный календарь для очереди 'Support-USA', с часовыми поясами, различающимися с очередью 'Support-Japan'. В OTRS можно создавать до 99 различных календарей.

7.1.1. Бизнес Время

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.

Заявки эскалируются, уведомления для эскалированных и ожидающих заявок отправляются и заявки разблокируются только в рабочее время.

7.1.2. Праздники с фиксированными датами

Празники, которые с каждым годом не приходятся на фиксированный день, такие как Новый год или четвертое Июля могут быть указаны в Отпускные дни или в соответствующем разделе календаря 1-9.

Заявки не будт эскалироваться, ни разблокироваться в сроки, определенные как "Специально зарезервированные дни".

Примечание

По умолчанию, OTRS работает по *Немецким* выходным.

7.1.3. TimeVacationDaysOneTime

Празники, такие как Пасха, которые ежегодно не имеют фиксированной даты могут быть указаны в Отпускные дни (One-Time) .

Заявки не будут эскалироваться и не будут разблокироваться в период времени определенный как TimeVacationDaysOneTime.

Примечание

OTRS поставляется без единого предустановленного One-Time-праздника. Это означает, что во время настройки системы OTRS, нужно самостоятельно добавить празники, такие как Пасха, День Благодарения.

7.2. Автоматические Разблокировки

Заблокированные заявки могут быть разблокированы системой автоматически. Эта опция может быть полезной, если, например, агент заблокировал заявки, которые нужно обработать, но не работал с ними по определенной причине, например был в отпуске. Чтобы убедится в том, что заблокированные заявки не будут забыты и их обработают другие агенты, автоматическая разблокировка разблокирует заблокированные заявки при истечении определенного времени.

Перейдя по ссылке настройки очереди для каждой заявки можно установить количество времени, при истечении которого она будет разблокирована. Модуль, `bin/otrs.UnlockTickets.pl`, который периодически вызывается как задание планировщика задач cron job, выполняет автоматическую разблокировку заявок.

Уведомления о разблокированных заявках отсылаются только тем агентам, которые добавили очередь с разблокированными заявками в "Мои очереди", и активировали в своих настройках функцию уведомлений о разблокированных заявоках.

Заявки будут заблокированы в том случае, если выполнятся следующие условия:

- Для очереди в которой хранится заявка определяется *время разблокировки* .
- Заявке присвоен статус *заблокирована*.
- Заявке присвоен статус *открыатая*.

Если агент добавляет в заявку новую статью - таймер разблокировки будет сброшен. Он может быть любым из следующих типов: *внешний-email*, *телефон*, *факс*, *смс*, или *внешнее примечание*.

Кроме этого, если последняя статья в заявке создана агентом, и клиент добавляет еще одну, либо через веб-интерфейс либо по электронной почте, таймер разблокировки также будет сброшен.

Последнее событие, которое будет сбрасывать таймер разблокировки, когда заявка передается другому агенту.

8. Настройка вывода в формате PDF

В этом разделе приводится настройка опций для экспорта данных из OTRS в PDF.

If you use the Print action from anywhere within the OTRS interface, it will generate a formatted PDF file. You can deactivate this by modifying the configuration parameter PDF to create HTML output instead.

Чтобы изменить внешний вид файлов, сгенерированных системой OTRS, можно создать свой собственный логотип и добавить его перейдя по ссылке PDF::LogoFile. Чтобы определить стандартный размер страниц генерируемых pdf-файлов (DIN-A4 or Letter) воспользуйтесь ссылкой PDF::PageSize. Чтобы указать максимальное количество страниц pdf-файла используйте ссылку PDF::MaxPage. Эта настройка может быть полезна когда пользователь ошибочно генерирует файл большого размера.

Для генерации pdf-файлов должны быть проинсталированы Perl-модули CPAN: PDF::API2 и Compress::Zlib. В большинстве дистрибутивов они доступны в виде пакетов и могут быть легко установлены с помощью соответсвтующего менедежера пакетов. В случае, если это не возможно, они должны быть установлены с помощью CPAN. За дополнительной информацией об установке Perl-модулей, обратитесь к главе "Установка Perl-модулей".

9. Stats-модуль

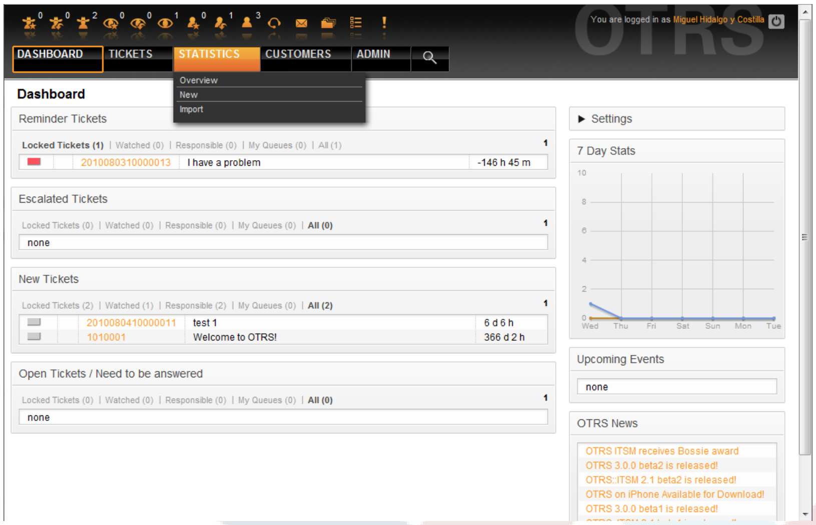
The OTRS stats module holds features to track operational statistics and generates custom reports associated with OTRS usage. The OTRS system uses the term "stat" generically to refer to a report presenting various indicators.

Правильная настройка модуля статистики OTRS связана с множеством различных условий. К ним относятся различные модули системы OTRS, подлежащие оценке, настройки прав доступа пользователей, признаки, которые должны быть рассчитаны и уровень их сложности, легкость настройки модуля статистики, скорость и эффективность вычислений, поддержка множества вариантов вывода информации.

Статистические элементы, такие как например файлы, которые дополняют функциональные возможности модуля статистики для конкретых требований, могут быть интегрированы для подсчета сложной статистики.

9.1. Обработка модуля агентом

When signed on as an agent, the navigation bar displays the link "Statistics", with various submenu options, as shown in Figure.



Dashboard

Reminder Tickets

Locked Tickets (1) | Watched (0) | Responsible (0) | My Queues (0) | All (1) 1

2010080310000013	I have a problem	-146 h 45 m

Escalated Tickets

Locked Tickets (0) | Watched (0) | Responsible (0) | My Queues (0) | All (0) 1

none

New Tickets

Locked Tickets (2) | Watched (1) | Responsible (2) | My Queues (0) | All (2) 1

2010080410000011	test 1	6 d 6 h
1010001	Welcome to OTRS!	366 d 2 h

Open Tickets / Need to be answered

Locked Tickets (0) | Watched (0) | Responsible (0) | My Queues (0) | All (0) 1

none

Upcoming Events

none

OTRS News

OTRS ITSM receives Bossie award
OTRS 3.0.0 beta2 is released!
OTRS::ITSM 2.1 beta2 is released!
OTRS on iPhone Available for Download!
OTRS 3.0.0 beta1 is released!

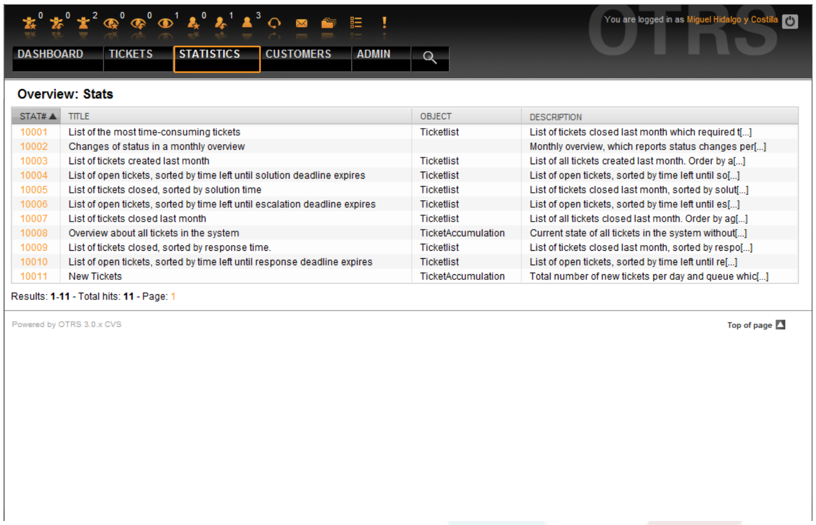
Рисунок: Опции меню Статистика.

Различные варианты представленные в меню статистики:

- *Overview*. Предоставляет список различных предварительно сконфигурированных отчетов.
- *Новая*. Требуются права rw.
- *Импорт*. Требуются rw-права.

9.1.1. Обзор

Selecting the "Statistics" link in the navigation bar, and then the submenu link "Overview", calls up the Overview screen. The Overview screen presents a list of all pre-configured reports the agent can use (see Figure below).



You are logged in as Miguel Hidalgo y Costilla

Overview: Stats

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

Results: **1-11** - Total hits: **11** - Page: 1

Powered by OTRS 3.0.x CVS

Top of page

Figure: Overview of the standard reports.

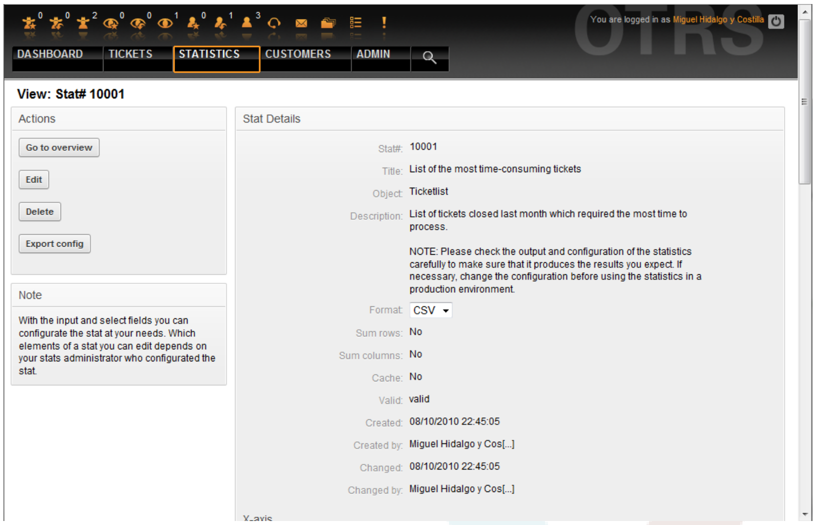
Для каждого из отчетов, перечисленных в Обзоре предоставлена следующая информация:

- *Stat#*. Уникальный номер отчета.
- *Заголовок*. Заголовок отчета.
- *Объект*. Объект, который используется для генерации статистики. В случае статической статистики не отображается ни один объект, поскольку в ее генерации динамические объекты не участвуют.
- *Описание*. Краткое описание отчета.

Когда модуль статистики установлен, он поставлятся с несколькими предустановленными отчетами, выборочно встроенными в систему. Они приведены в виде списка на странице "Обзор". Если список слишком большой и не помещается на одну страницу, у агента есть возможность просматривать другие страницы. Список отчетов может быть отсортирован по своему вкусу, путем нажатия на желаемый заголовок столбца в списке. Чтобы создать конкретный отчет, нажмите на соответствующие stat-номер, связанный с отчетом. В результате откроется интерфейс просмотра отчета.

9.1.2. Создание и просмотр отчетов

The view user interface provides the stat's configuration settings (see Figure below).



View: Stat# 10001

Actions

Note

With the input and select fields you can configurate the stat at your needs. Which elements of a stat you can edit depends on your stats administrator who configurated the stat.

Stat Details

Stat#: 10001

Title: List of the most time-consuming tickets

Object: Ticketlist

Description: List of tickets closed last month which required the most time to process.

NOTE: Please check the output and configuration of the statistics carefully to make sure that it produces the results you expect. If necessary, change the configuration before using the statistics in a production environment.

Format: CSV

Sum rows: No

Sum columns: No

Cache: No

Valid: valid

Created: 08/10/2010 22:45:05

Created by: Miguel Hidalgo y Cos[...]

Changed: 08/10/2010 22:45:05

Changed by: Miguel Hidalgo y Cos[...]

X-axis

Figure: Viewing a specific report.

Параметры конфигурации для конкретного отчета могут быть установлены путем выбора ряда возможных опций. Как автор отчета так и другие пользователи с соответсвтующими правами доступа могут создавать настройки.

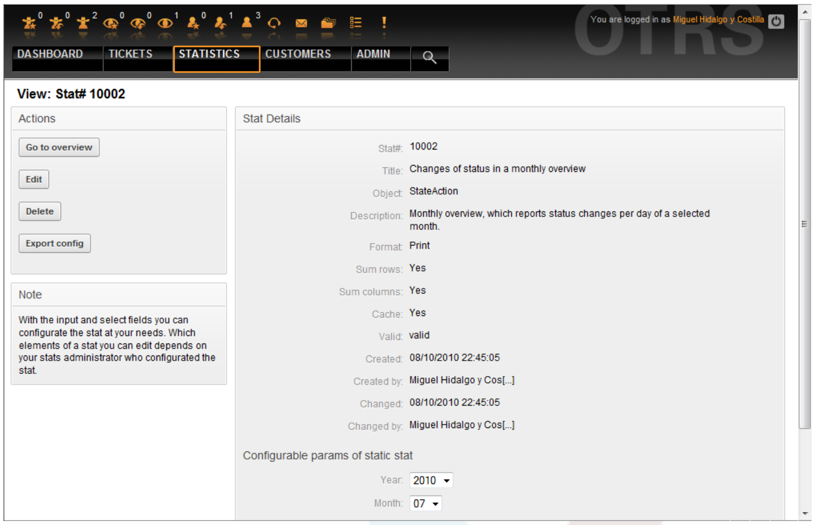
На странице отображается следующее:

- Возможные действия:
  - *Перейти к обзору*. Ссылка, ведущая обратно к Просмотру списка отчетов.
  - *Редактировать*. Редактировать имеющуюся структуру отчета (требуются права для чтения и записи, rw).
  - *Удалить*. Удалить текущий отчет (требуются права для чтения и записи, rw).
  - *Экспортировать настройки*. Экспортировать настройки отчета с помощью загрузки файла (требуются права для чтения и записи, rw).

    Традиционно: функции Экспорт и Импорт позволяют создавать и тестировать отчеты та тестовых системах с их легкой дальнейшей интеграцией в производственную систему.
- Подробности отчета:
  - *Отчет#*. Номер отчета.
  - *Заголовок*. Заголовок отчета.
  - *Объект*. Объект, который используется для создания отчета.
  - *Описание*. Описание цели отчета.
  - *Формат*. Выходной формат отчета. В зависимости от конфигурации, можно выбрать один из следующих форматов вывода:
    - CSV.
    - Печать.

- Графические линии.
- Столбцовая диаграмма.
- Диаграмма.
- Точечная диаграмма.
- Точечно-линейная диаграмма.
- Диаграмма
- Круговая "пирожок"-диаграмма.

- *Размер в пикселях*. Размер графиков/диаграмм в пикселях. Эта опция доступна только в том случае, если в настройках отчета разрешены графики. Все часто используемые графические размеры настраиваются OTRS-администратором в SysConfig. Затем, во время настройки отчета, агент может выбрать необходимый формат из всех доступных.
- *Количество строк*. Указывает когда отчет заменяется колонкой, чьи ячейки содержат сумму соответствующих строк.
- *Количество колонок*. Указывает когда отчет заменяется строкой, чьи ячейки содержат сумму соответствующих колонок.
- *Кэш*. Указывает когда сгенерированный отчет кэшируется в файловой системе.
- *Действительный*. Может быть установлен в значение "недействительный" если по какой то причине отчет временно не должен запускаться. Кнопка "Старт" в нижней части правой панели больше не отображается. Таким образом отчет не может быть создан.
- *Создан*. Время создания отчета.
- *Создан (кем)*. Имя агента, который создал отчет.
- *Изменен*. Время последней модификации отчета.
- *Изменен кем*. Имя агента, который последним внес изменения в отчет.
- *X-axis*. Используя эту функцию, агент может переключать x и y axes (только в том случае, если эта опция активирована OTRS-администратором).
- Общая информация сопровождается информацией о самом отчете. Есть два различных шаблона просмотра отчета (или статистики):
    - *Static stat view*. Static report generators can be integrated into the stats module (see Figure below).



View: Stat# 10002

Actions

Note

With the input and select fields you can configurate the stat at your needs. Which elements of a stat you can edit depends on your stats administrator who configurated the stat.

Stat Details

Stat#: 10002
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
Created: 08/10/2010 22:45:05
Created by: Miguel Hidalgo y Cos[...]
Changed: 08/10/2010 22:45:05
Changed by: Miguel Hidalgo y Cos[...]

Configurable params of static stat

Year: 2010
Month: 07

Figure: Viewing a static report.

- *Dynamic stat view* (see Figure above). They can be displayed in two different ways:
  - *Неизменяемые настройки*. Создатель отчета не имеет прав для изменения этих полей.
  - *Изменяемые настройки*. Параметры конфигурации таких отчетов могут быть изменены агентом.

Нажатие кнопки "Пуск" (в нижней части экрана) является последним шагом при создании отчета. Есть две возможные причины, когда эта кнопка не будет отображаться:

1. Отчет был установлен недействительным и, таким образом отключен.
2. Отчет был настроен не аккуратно и, следовательно, не может выполнятся. В этом случае необходимую информацию можно найти в разделе уведомлений OTRS (внизу, под навигационной панелью).

Если настройки на странице Просмотр неверны, эта страница отображается опять, после нажатия кнопки "Пуск" и информация о введеных неверных данных будет выведена в разделе уведомлений.

9.1.3. Редактировать / Создать новый

Агенты, которые имеют права на запись, могут редактировать настройки существующего отчета, запустив пользовательский веб-интерфейс модуля статистики. В качестве альтернативы они могут создать новый отчет. Соответствующие страницы могут быть запущены следующим образом:

1. Редактировать: Используя кнопку "Редактировать" в шаблоне просмотра статистики.
2. Новый: Используя ссылку "Новый" в меню Статистика навигационной панели, или кнопку "Добавить" на странице Просмотр.

Статистика редактируется с помощью мастера в четыре этапа:

1. Общие технические условия.

2. Определение элементов для X-axis.

3. Спецификация значений ряда.

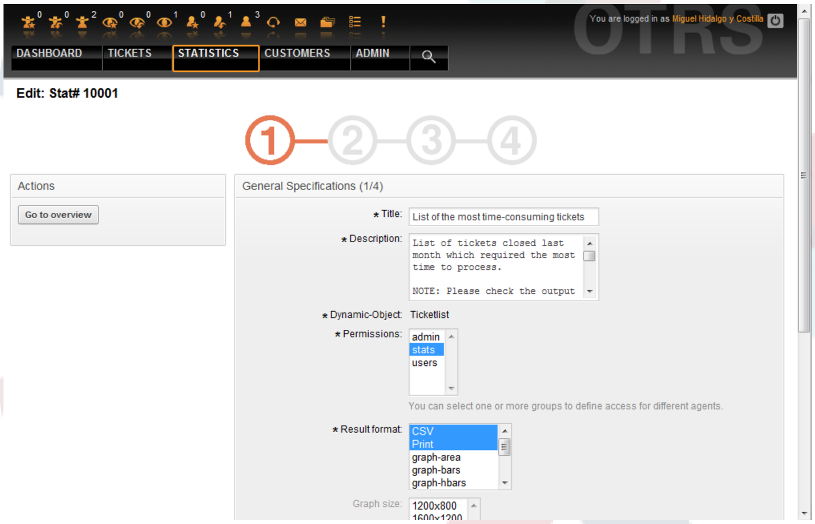
4. Выбор ограничений для отчета.

Шаги с 2-4 необходимы только для создания отчетов с динамической статистикой. Для статической статистики требуется только общая информация (пункт 1).

Информация о том, как обрабатывать страницу приводится на каждом из этих экранов, ниже панели "Действия" панели Подсказок.

Если введены неправильные данные, предыдущий пользовательский интерфейс отображатеся заново, на этот раз уже с информацией о неправильных входных данных. Эта информация отображается в OTRS в разделе уведомлений. Следующий пользовательский веб-интерфейс для ввода отображатеся только в том случае, если все поля формы заполнены правильно.

1. *General specifications*. It is the first page of the Edit wizard (see Figure below).



Edit: Stat# 10001

General Specifications (1/4)

* Title: List of the most time-consuming tickets

* Description:

```
List of tickets closed last
month which required the most
time to process.

NOTE: Please check the output
```

* Dynamic-Object: Ticketlist

* Permissions: admin, stats (selected), users

You can select one or more groups to define access for different agents.

* Result format: CSV (selected), Print (selected), graph-area, graph-bars, graph-hbars

Graph size: 1200x800, 1600x1200

Figure: Editing the general specifications of a report.

In the screen showed in Figure, there are a great number of common specifications and settings that can be edited:

- *Заголовок*. Должны в сжатой форме отражать назначение (цель).

- *Описание*. Более подробная информация об отчете, типах конфигурационных параметров и т.д.

- *Динамический объект*. Если установленная система OTRS предоставляет различные динамические объекты, один из них может быть выбран. Объекты отвечают требованиям отдельных модулей.

- *Static file*. Обычно эта возможность не отображается до тех пор, пока отображаются статические файлы, назначеные отчету. Если "Статический файл" отображается, позже, очень важно поставить галочку в поле и выбрать режим генерации (динамический с динамический объектом или статический с файлом). Если выбран статический файл, пользовательский веб-интерфейс 2-4 не отображается как статический файл содержащий все необходимые настройки.

- *Настройки прав доступа*. Содействие ограничению групп (и, следовательно, агентов), которые позже могут просматривать и генерировать предварительные отчеты. Таким образом, различные отчеты могут быть выделена для различных отделов и рабочих групп, которые в них нуждаются. Можно выделить одину статистику для различных групп.

  Пример 1: Выбрана "stats" группа. Отчет доступен для просмотра для всех пользователей имеющих по крайней мере права на группу "stats . Этот доступ доступен по умолчанию.

  Пример 2: Была выбрана группа под названием "продажи". Все пользователи, имеющие права ro (read only) к этой группе могут выдеть статистику в режиме просмотра э генерировать ее. Однако отчет не будет доступен для просмотра другим пользователям.

- *Формат*. Формат отображения статистики. В зависимости от конфигурации, могут быть выбраны следующие один или несколько форматов:

  - CSV.
  - Печать.
  - graph-lines.
  - graph-bars.
  - graph-hbars.
  - graph-points.
  - graph-lines-points.
  - graph-area.
  - graph-pie.

- *Размер графика*. Выберите размер диаграммы в пикселях. Этот выбор необходим только в том случае, если графический формат вывода был выбран в "Формат". Все графические размеры, которые можно использовать определяются в OTRS в SysConfig. При настройке отчета, агент может предварительно выбрать все соответствующие форматы.

- *Количество строк*. Указывает когда отчет заменяется колонкой, чьи ячейки содержат сумму соответствующих строк.

- *Сумма столбцов*. Указывает когда отчет заменяется строкой, чье ячейки содержат сумму соответствующих строк.

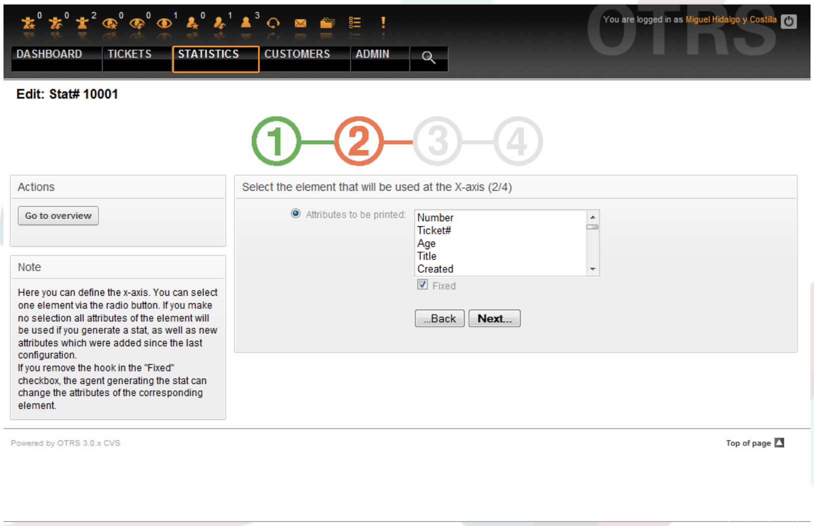
- *Кэш*. Указывает на то, что созданный отчет должен кэшироваться в файловой системе. Это позволяет существенно экономить вычислительные мощности и время, но должно использоваться только в том случае если содержание отчета больше не будет изменятся.

Кэширование автоматически прекращается, если отчет не содержит временных обозначений, или когда временные обозначения указывают на будущее.

В случае редактирования закэшированного отчета удаляются все закэшированные данные.

- *Действительный*. Устанавливается в значение "недействительный" если ранее сконфигурированный отчет по какой то причине не должен выполнятся. Кнопка "Статистика" также не отображатеся в правом нижнем углу блока. Отчет больше не генерируется.

2. *Определение элементов для X-axis*. Это конфигурация для элементов, которые используются для описания X-axis или, если используются таблицы, имя столбца применяется для X-axis (см. Рисунок ниже).



Edit: Stat# 10001

Select the element that will be used at the X-axis (2/4)

● Attributes to be printed: Number, Ticket#, Age, Title, Created

☑ Fixed

Here you can define the x-axis. You can select one element via the radio button. If you make no selection all attributes of the element will be used if you generate a stat, as well as new attributes which were added since the last configuration.
If you remove the hook in the "Fixed" checkbox, the agent generating the stat can change the attributes of the corresponding element.

Рисунок: Определение элементов для X-axis.

Во-первых, элемент выбирается с помощью опционального поля. Потом нужно выбрать два или более атрибутов элемента. Если выбраных атрибутов нету, используются все атрибуты, в том числе и те, которые были добавлены после конфигурации отчета.

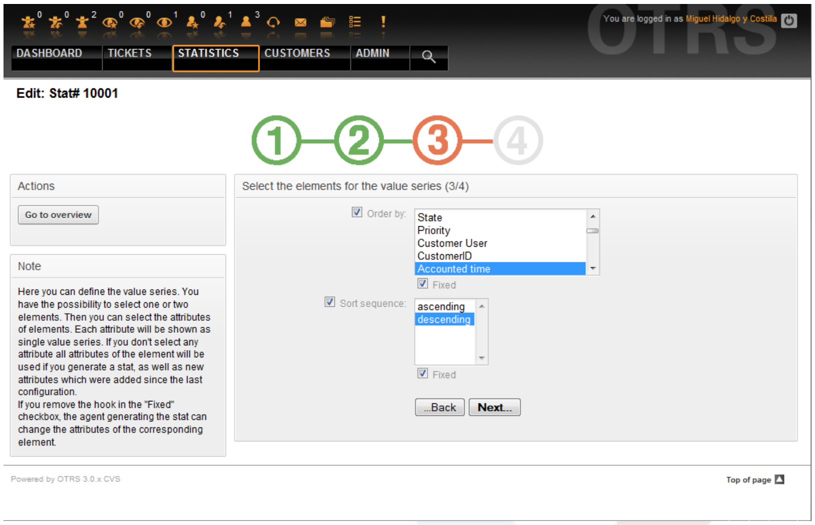
Если отключена настройка "Фиксированный", агент, генерирующий отчет, может изменить атрибуты соответствующего элемента в меню "Вид" пользовательского веб-интерфейса.

Элементы времени отличаются от периодов времени. Тип и число элементов зависит от использующихся динамических обьектов и сильно зависит от этого.

Если все входные данные введены правильно, нажатие кнопки "Следующий шаг" открывает форму "Серия значений". Кроме этого, есть возможность заново вернуться к редактированию предыдуших разделов.

3. *Спецификация серии значений*.

In the third step of the report configuration, the value series are defined (see Figure below). They will later form the individual graphs or the various series within a tabular view.



Edit: Stat# 10001

Note

Here you can define the value series. You have the possibility to select one or two elements. Then you can select the attributes of elements. Each attribute will be shown as single value series. If you don't select any attribute all attributes of the element will be used if you generate a stat, as well as new attributes which were added since the last configuration.
If you remove the hook in the "Fixed" checkbox, the agent generating the stat can change the attributes of the corresponding element.

Select the elements for the value series (3/4)

☑ Order by: State, Priority, Customer User, CustomerID, Accounted time (selected)

☑ Fixed

☑ Sort sequence: ascending, descending (selected)

☑ Fixed

Рисунок: Определение значений серии.

Если выбран определенный элемент, каждый выбранный атрибут будет соответствовать значеню серии (см. ниже Пример 19-1).

Пример 4.19. Определение значения серии - один из элементов

Элемент Очередь:

- Серия значений 1 = Raw
- Значение серии 2 = Junk
- ....

Если для формирования значения ряда, выбрано два элемента, каждый выбраный атрибут первого элемента комбинируется с атрибутом второго элемента (см. ниже Рисунок 19-2).

Пример 4.20. Определение значения серии - один из элементов

Элемент 1 очередь, Элемент 2 статус:

- Значение создания 1 = Raw - открыть
- Серия значений 2 = Raw - успешно закрыта
- Серия значений 3 = Junk - открыто
- Серия значений 4 = Junk - успешно закрыто

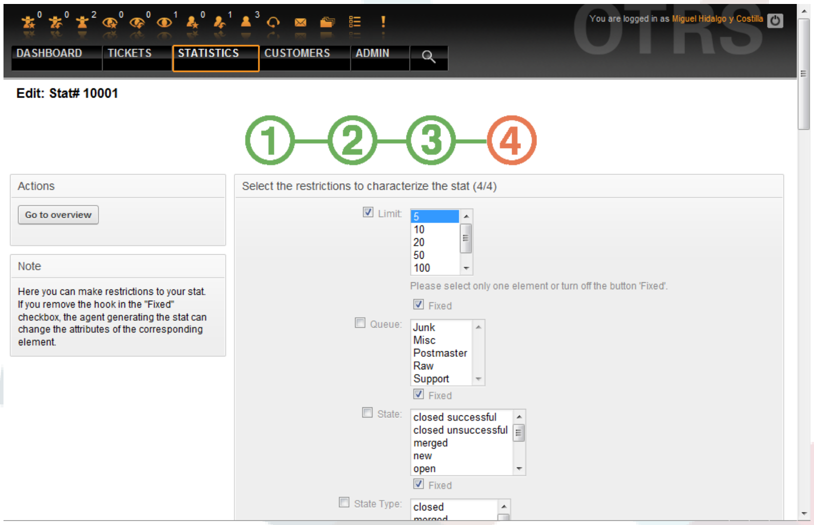
Выбор из трех или более элементов не допускается.

Кроме этого эти условия распостраняются и на выбор атрибутов и флажок "Основные" как для "X-axis":

- Если для элемента не выбрано никаких атрибутов, то используются все атрибуты, включая и те, которые были добавлены после настройки отчета.

- Если параметр "Фиксированный" отключен, агент, который создает отчет может изменять атрибуты соответствующих элементов.

4. *Установка ограничений отчета*. Это четвертый и последний этап конфигурации (см. Рисунок ниже). Ограничения служат для ограничения результатов выбранных критериев. В ряде случаев не должно быть никаких ограничений.



Edit: Stat# 10001

Here you can make restrictions to your stat. If you remove the hook in the "Fixed" checkbox, the agent generating the stat can change the attributes of the corresponding element.

Select the restrictions to characterize the stat (4/4)

☑ Limit: 5, 10, 20, 50, 100

Please select only one element or turn off the button 'Fixed'.

☑ Fixed

□ Queue: Junk, Misc, Postmaster, Raw, Support

☑ Fixed

□ State: closed successful, closed unsuccessful, merged, new, open

☑ Fixed

□ State Type: closed, merged

Рисунок: Определение ограничений.

После установки всех ограничей, конфигурация отчета завершается нажатием кнопки "Готово".

9.1.4. Импорт

Для доступа к пользовательскому интерфейсу Импорта (см. Рисунок ниже) перейдите по ссылке "Статистика", а затем выберите "Импорт". Альтернативный способ - на странице "Обзор" нажать кнопку Импорт, это приведет к тому же результату, но для этого действия требуются права для чтения и записи "RW".

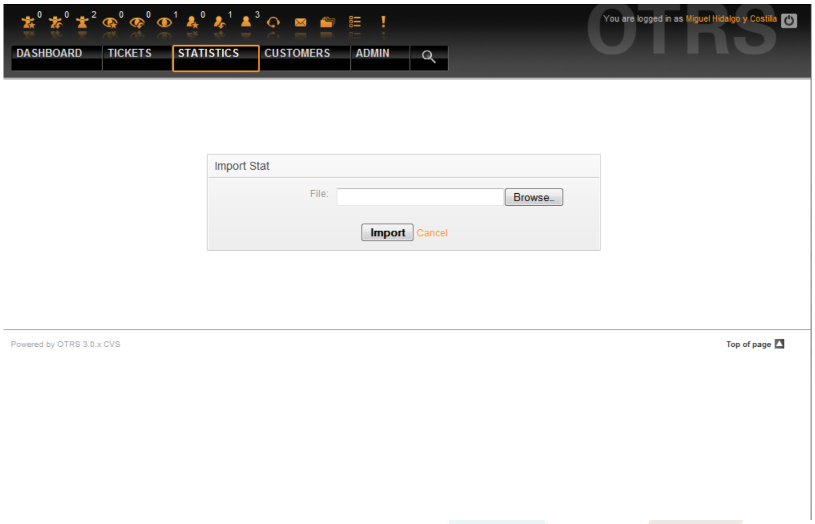


Рисунок: Пользовательский веб-интерфейс для Импорта.

Очень удобная функция, значительно облегчает импорт статистики, особенно в сочетании с возможностями модуля экспорта. Статистику можно создать и протестировать на тестовых системах, а затем импортировать в реальную систему.

Импорт осуществляется путем простой загрузки файла. После этого, для импорта статистики, автоматически открывается пользовательский интерфейс "Вид".

9.2. Администрирование модуля статистики OTRS администратором.

Этот раздел предоставляет информацию о задачах и функциях администратора OTRS, который работает с модулем статистики.

9.2.1. Настройки полномочий, Групп и Ролей

Нет новых очередей и/или групп, созданых при установке модуля статистики.

В конфигурации по умолчанию модуль регистрации дает всем агентам группы "статистика" права доступа к модулю статистики.

Доступ в соответствии с разрешенными настройками:

- *rw*. Разрешает настройку статистики и отчетов.
- *ro*. Разрешает генерацию предварительно сконфигурированной статистики и отчетов.

OTRS-администратор самостоятельно определяет когда агенты с правами для генерации предварительно настроенных отчетов выделяются в ro-права для группы "статистика", или если их соответствующие группы добавляются в модуль регистрации в SysConfig.

9.2.2. SysConfig

Группы SysConfig Framework:Core::Stats, Framework:Core::Stats::Graph and Framework:Frontend::Agent::Stats содержат все параметры конфигурации базового набора из модуля статистики. Больше того, конфигурационный параметр $Self->{'Frontend::Module'}->{'AgentStats'} управляют организацией и регистрацией модулеи и иконкой в модуле статистики.

9.3. Администрирование модуля статистики системным администратором

Как правило, системного администратора не требуется для эксплуатации, настройки и обслуживания модуля статистики. Однако, немного справочной информации дано для системного администартора в этой точке.

Примечание

Пути к файлам см. в подкаталогах домашнего каталога OTRS (в большинстве случаев `/opt/otrs`).

9.3.1. Таблица базы данных

Включены все конфигурационные параметры отчета и управляются в XML и, следовательно, хранятся в таблице базы данных "xml_storage". Другие модули, конкент которых хранится в xml-формате также используют эту таблицу.

9.3.2. Список файлов

Следующие файлы необходимы для коректной работы модуля статистики:

- `Kernel/System/Stats.pm`
- `Kernel/Modules/AgentStats.pm`
- `Kernel/System/CSV.pm`
- `Kernel/Output/HTML/Standard/AgentStatsOverview.dtl`
- `Kernel/Output/HTML/Standard/AgentStatsDelete.dtl`
- `Kernel/Output/HTML/Standard/AgentStatsEditSpecification.dtl`
- `Kernel/Output/HTML/Standard/AgentStatsEditRestrictions.dtl`
- `Kernel/Output/HTML/Standard/AgentStatsEditXaxis.dtl`
- `Kernel/Output/HTML/Standard/AgentStatsEditValueSeries.dtl`
- `Kernel/Output/HTML/Standard/AgentStatsImport.dtl`
- `Kernel/Output/HTML/Standard/AgentStatsPrint.dtl`
- `Kernel/Output/HTML/Standard/AgentStatsView.dtl`
- `Kernel/System/Stats/Dynamic/Ticket.pm`
- `bin/otrs.GenerateStats.pl`

9.3.3. Кэширование

Результаты статистики должны быть сохранены в кэше или не могут быть установлены в конфигурации. Кэшированые результаты отчетов хранятся в файлах в каталоге `var/tmp` установленной системы OTRS (в большинстве случаев `/opt/otrs/var/tmp`).

Кэшированная статистика может быть распознана с помощью префикса "Stats".

В случае потери данных системе не будет принесено серьезного ущерба. При вызове модуля статистики в следующий раз, в случае если он не обнаружит файл, отчет сгенерируется заново. Конечно, это может занять немного больше времени.

9.3.4. otrs.GenerateStats.pl

Этот файл хранится в директории `bin`. Он позволяет генерировать статистику прямо из командрной строки.

В качестве примера смотрите вызов Сценария из командной сторки, который приведен ниже.

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

Сценарий: Создание отчета из командной строки.

Отчет из конфигурации статистики "Stat# 10004" генерируется и хранится в csv-формате в директории `/output/dir`.

Сгенерированный отчет также быть отправлен по электронной почте. Более подробную информацию можно получить выполнив команду из Сценария приведенног ниже.

```
bin> perl otrs.GenerateStats.pl --help
```

Сценарий: Получение информации о файле otrs.GenerateStats.pl.

9.3.5. Автоматическое создание статистики с помощью планировщика заданий - Cronjob

Очевидным является тот факт, что в действительности, нету необходимости генерировать отчеты вручную, из командной строки, так как модудь статистики имеет удобный графический веб-интерфейс пользователя. Однако, создание отчетов вручную имеет смысл в сочетании с Cronjob.

Представьте себе следующую ситуацию: Каждый первый день месяца, руководители отделов хотят получать отчет за последний месяц. Объединив cronjob с командой запуска сценария в командной строке, статистика может отправляться им по электронной почте автоматически.

9.3.6. Статическая статистика

Модуль статистики облегчает генерацию статической статистики. Для каждой статической статистики создается отдельный файл, в котором точто определено его содержание.

Таким образом может быть сгенерирована очень сложная статистика. Главным недостатком является то, что этот способ есть не очень гибким.

Файлы хранятся в директории `Kernel/System/Stats/Static/`.

9.3.7. Использование устаревших данных статической статистики

Прежние версии OTRS 1.3 и 2.0 уже позволяли генерировать отчеты. Различные отчеты для OTRS версии 1.3 и 2.0 которые были специально разработаны для удовлетворения потребностей клиентов могут быть использованы в более позних версиях OTRS.

Файлы должны быть просто перемещены из Kernel/System/Stats/ в Kernel/System/Stats/Static/. Кроме этого имя пакета соответствующего сценария должен быть заменен на "::Static".

В следующем примере показано как изменяется первый путь.

```
    package Kernel::System::Stats::AccountedTime;
```

```
    package Kernel::System::Stats::Static::AccountedTime;
```

9.3.8. Статистика по умолчанию

"Не всегда нужно изобретать велосипед..."

Модуль статистики предоставляет различные предустановленные отчеты. Предустановленные отчеты, которые могут заинтересовать всех OTRS пользователей можно в будущем добавить в модуль статистики. Предустановленные отчеты хранится в модуле статистики в xml-формате в директории scripts/test/sample/.

10. Динамические Поля

10.1. Введение

A dynamic field is a special kind of field in OTRS, created to extend the information stored on a ticket or article. These fields are not fixed in the system and they can appear only in specific screens, they can be mandatory or not, and their representation in the screens depends on the field type defined at their creation time according to the data to be held by the field. For example, there are fields to hold a text, a date, a selection of items, etc.

Dynamic fields are the evolution of TicketFreeText TicketFreeKey TicketFreeTime, ArticleFreeText and ArticleFreeKey fields that where commonly used in OTRS 3.0 and before. The limitation of these "Free Fields" was that they can be defined up to 16 (text or dropdown) fields and 6 time fields for a ticket and 3 (text or dropdown) fields for each article only, not more.

Now with dynamic fields the limitation in the number of fields per ticket or article is removed, you can create as many dynamic fields you like either for ticket or articles. And beyond that, the framework behind the dynamic fields is prepared to handle custom fields for other objects rather than just ticket and articles.

This new framework that handles the dynamic fields is build using a modular approach, where each kind of dynamic field can be seen as a plug-in module for the framework. This means that the variety of dynamic fields can be easily extended by public OTRS modules, OTRS Feature Add-ons, OTRS custom developments, and other custom developments.

В этом релизе доступны следующие типы динамических полей:

- Text (одна строка текста)
- Textarea (несколько строк текста)
- Checkbox
- Выпадающий список (единственный выбор, несколько значений)

- Multiselect (множественный выбор, несколько значений)
- Date
- Date / Time

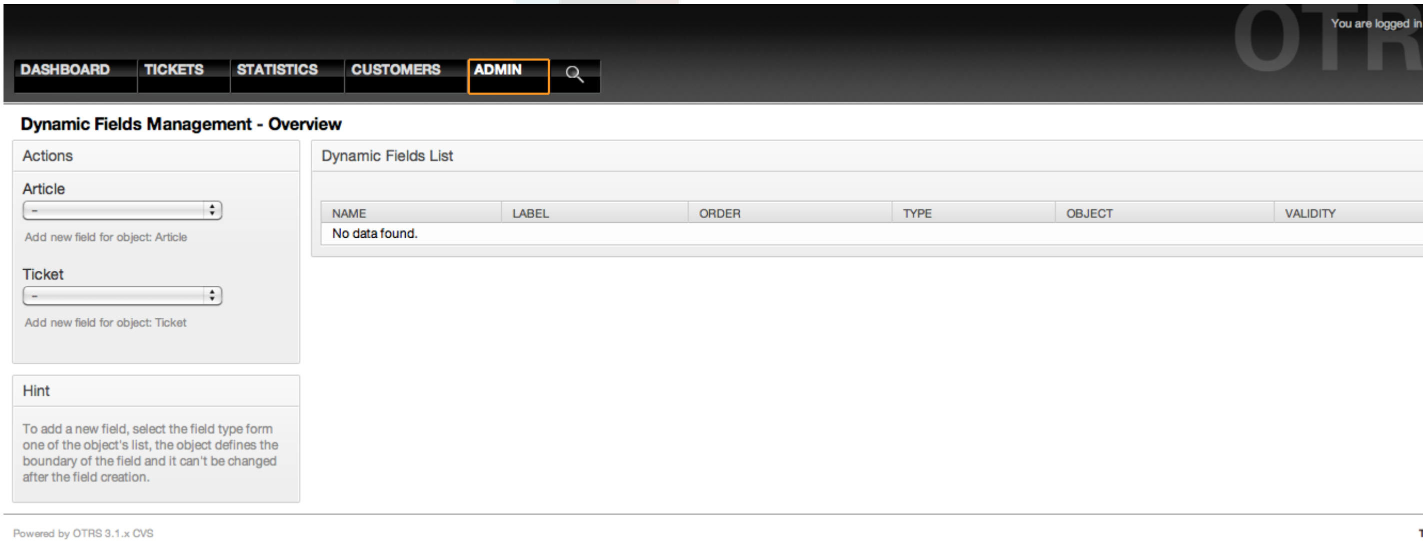
10.2. Настройка Системы

By default, a clean installation of OTRS does not include any dynamic fields. If you plan to use such fields in tickets or articles you need to create dynamic fields.

The configuration of a dynamic field is split in two parts, to add a new dynamic field or manage an existing one you need to navigate into the "Admin" panel in the "Dynamic Fields" link. To show, show as mandatory or hide a dynamic field in one screen you need to change the OTRS settings in the "SysConfig" screen.

10.2.1. Добавление Динамического Поля

Click on the "Admin" button located in the navigation bar, then click on the "Dynamic Field" link inside "Ticket Settings" box located in the lower center of the screen. The dynamic fields overview will display as follows:



Dynamic Fields Management - Overview

Actions

Article

-

Add new field for object: Article

Ticket

-

Add new field for object: Ticket

Hint

To add a new field, select the field type form one of the object's list, the object defines the boundary of the field and it can't be changed after the field creation.

Dynamic Fields List

NAME	LABEL	ORDER	TYPE	OBJECT	VALIDITY
No data found.					

Powered by OTRS 3.1.x CVS

Рисунок: Страница просмотра Динамических полей, пустая

Notice that this screen will change as you add more dynamic fields to list all created dynamic fields. This screen might already have some fields if the installation was updated from an older version of OTRS.

The Actions in the side bar at the left of the screen describes two possibilities: Article and Ticket, each one has it's own dropdown selection of dynamic fields.

Примечание

The installation of an OTRS package could add more objects to the Action side bar.

Общая процедура для создания динамического поля:

- Кликните на объект выбраного вами динамического поля в выпадающем списке на боковой панели действий.
- Выберите в списке и нажмите на тип динамического поля, которое вы хотите добавить.
- Настройка
- Сохранить.

Диалог настройки динамических полей разделен на две части, верхняя часть является общей для всех полей, а нижняя часть может различаться в зависимости от типа динамического поля.

Общие настройки динамического поля:

- Название: Обязательное поле, уникальное, только буквы и числа в качестве допустимых символов.

  This is the internal name of the field, used for example to show or hide a field in one screen. Any modification of a field name (not recommended) requires a manual update of the "SysConfig" settings where the field is referenciated.

- Метка: Обязательное.

  This is the field name to be displayed on the screens, it supports translations.

  ### Примечание

  Перевод метки должен быть добавлен в файл перевода вручную.

- Порядок Полей: Обязательно.

  Defines the relative order in which the field will be displayed on the screen, by default each new field has the last position, a change in this setting will affect the order of the other created dynamic fields.

- Срок действия: Обязательный.

  Не действительное динамическое поле не будет отображаться на любом экране, независимо от того какое значение установлено в конфигурации.

- Тип поля: Обязательное, Только для чтения.

  Показывает текущий выбранный тип поля.

- Тип Объекта: Обязательный, Только для чтения.

  Отображает область действия для поля.

Примечание

Для иллюстрации каждого конкретного параметра настройки в нашем примере будет добавлено несколько полей. Ссылки на эти поля будут упоминаться в последующих разделах.

В следующих примерах все динамические поля будут созданы для объекта Заявка, если необходимо создать динамическое поле для объекта Статья, просто выберите поле из выпадающего списка.

Таблица 4.6. В систему будут добавлены следующие поля:

Название	Метка	Type
Поле1	Мое Поле 1	Text
Поле2	Мое Поле 2	Textarea
Поле3	Мое Поле 3	Checkbox
Поле4	Мое Поле 4	Dropdown
Поле5	Мое Поле 5	Multiselect (множественный выбор)
Field6	Мое Поле 5	Date
Поле7	Мое Поле 6	Date / Time

10.2.2. Настройка Динамического Поля Text

Динамические текстовые поля используются для хранения простой строки.

Настройки динамического поля Text:

- Значение по умолчанию: Необязательное.

  This is the value to be shown by default on the edit screens (like New Phone Ticket or Ticket Compose).

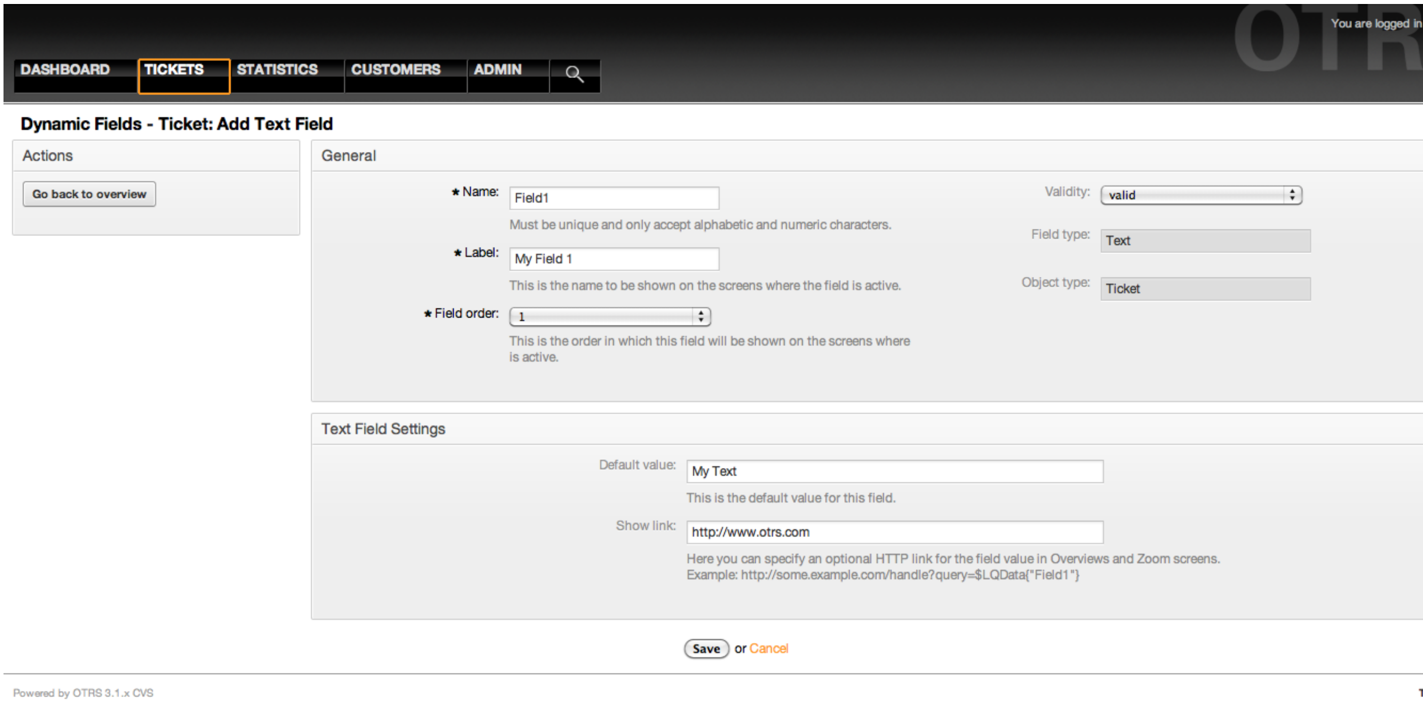
- Показать ссылку: Необязательное.

  Если установлено, то значение поля будет преобразовано в кликабельную ссылку, которая будет отображаться на экране.

  Например, если "Показать ссылку" установленно в "http://www.otrs.com", нажатие на заполненное значение приведет к открытию веб-страницы OTRS.

Примечание

The use of $LQData{"NameX"} in the Set link value, where NameX is the name of the field will add the field value as part of the link reference.



Dynamic Fields - Ticket: Add Text Field

Actions

General

★ Name: Field1

Must be unique and only accept alphabetic and numeric characters.

★ Label: My Field 1

This is the name to be shown on the screens where the field is active.

★ Field order: 1

This is the order in which this field will be shown on the screens where is active.

Validity: valid

Field type: Text

Object type: Ticket

Text Field Settings

Default value: My Text

This is the default value for this field.

Show link: http://www.otrs.com

Here you can specify an optional HTTP link for the field value in Overviews and Zoom screens. Example: http://some.example.com/handle?query=$LQData{"Field1"}

Powered by OTRS 3.1.x CVS

Рисунок: Диалоговое окно настройки Динамического поля Text.

10.2.3. Настройка Динамического Поля Textarea

Динамическое поле Textarea используется для хранения строк, состоящих из нескольких линий.

Настройки динамического поля Textarea:

- Количество строк: Необязательное, Целое.

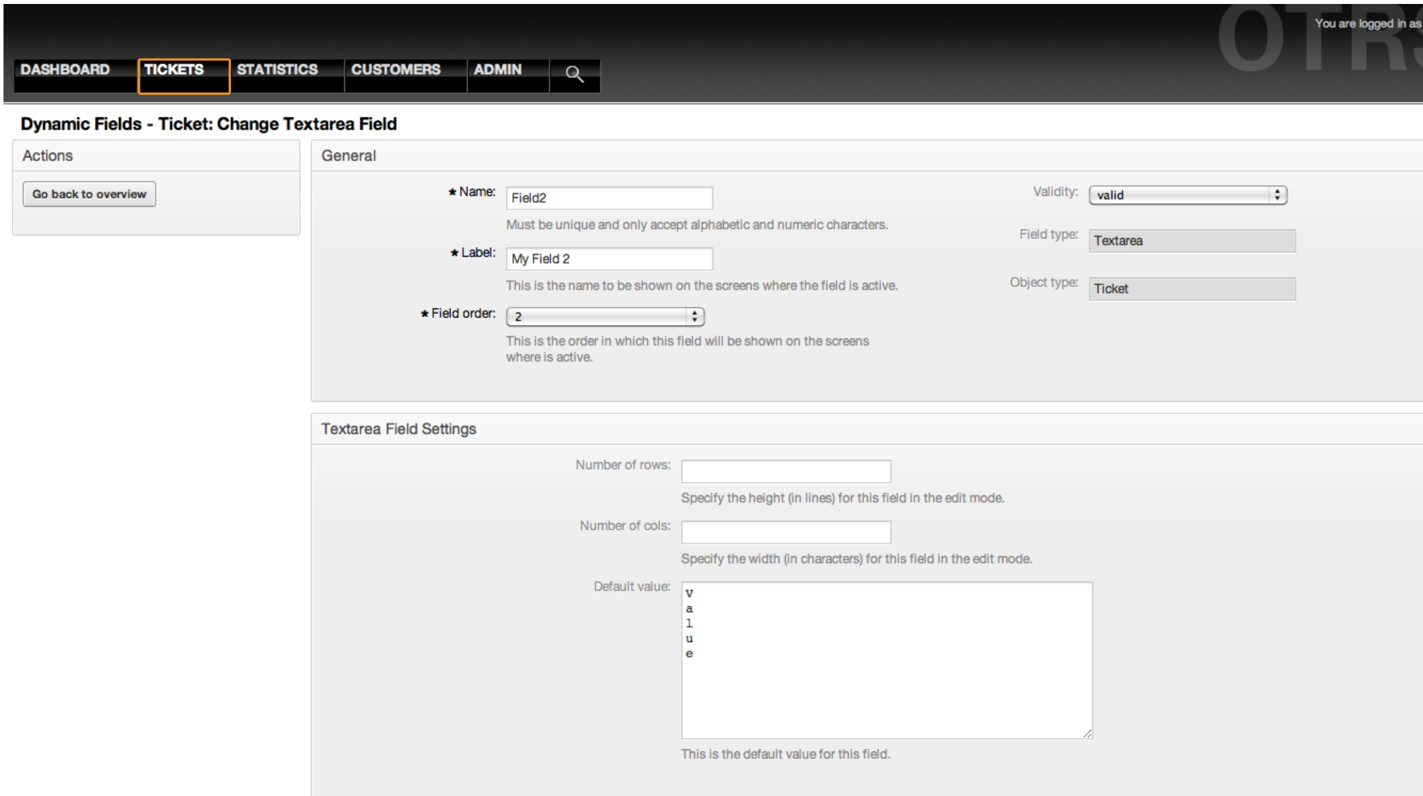
  Используется для определения высоты поля на странице редактирования.

- Количество столбцов: Необязательное, Целое.

  This value is used to define the width of the field in the edit screens.

- Значение по умолчанию: Необязательное.

  Это значение будет отображаться по умолчанию (может быть текстом, состоящим из нескольких строк).



Dynamic Fields - Ticket: Change Textarea Field

General

- ★ Name: Field2
  Must be unique and only accept alphabetic and numeric characters.
- ★ Label: My Field 2
  This is the name to be shown on the screens where the field is active.
- ★ Field order: 2
  This is the order in which this field will be shown on the screens where is active.
- Validity: valid
- Field type: Textarea
- Object type: Ticket

Textarea Field Settings

- Number of rows:
  Specify the height (in lines) for this field in the edit mode.
- Number of cols:
  Specify the width (in characters) for this field in the edit mode.
- Default value:

```
v
a
l
u
e
```

  This is the default value for this field.

Рисунок: Окно настройки Динамического Поля Textarea.

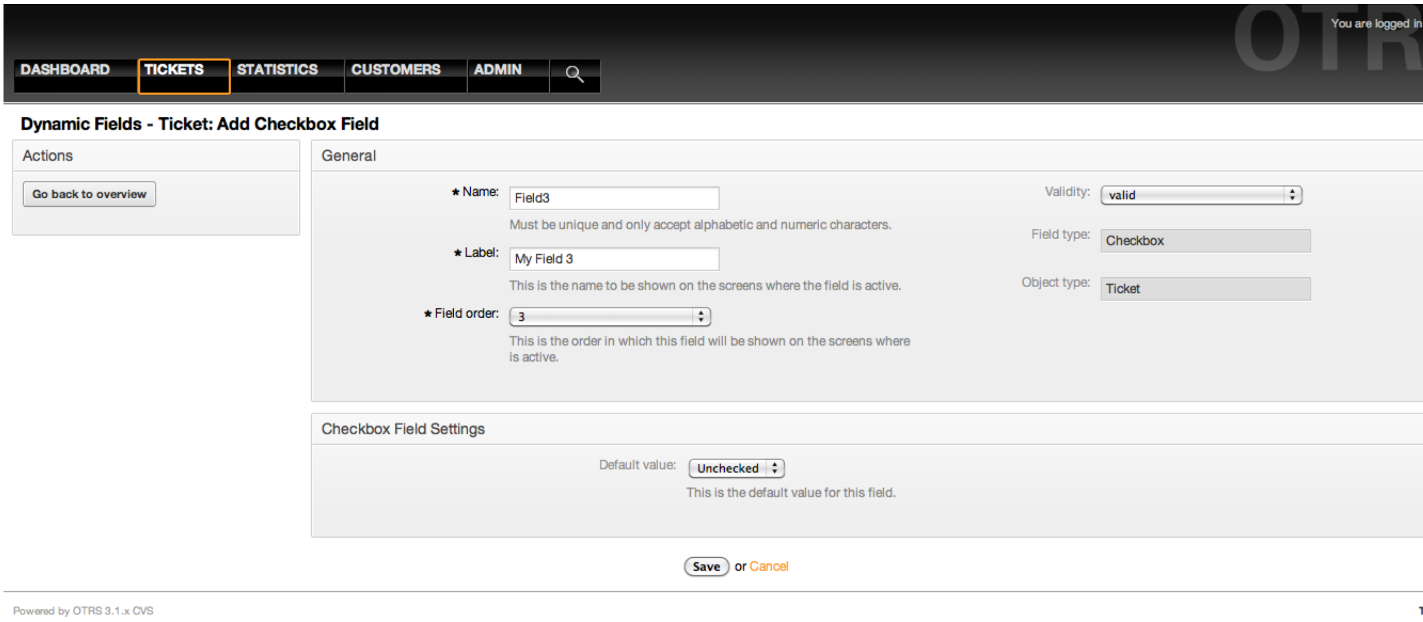
10.2.4. Конфигурация Динамического Поля Checkbox

Динамическое поле Checkbox используется для хранения значений true или false, изображено как выбранныи или не выбранный флажок.

Настройки динамического поля Checkbox:

- Значение по умолчанию: Обязательное.

  This is the value to be shown by default on the edit screens (like New Phone Ticket or Ticket Compose), the default value for this field is closed selection which can be Checked or Unchecked.



Dynamic Fields - Ticket: Add Checkbox Field

★ Name: Field3

Must be unique and only accept alphabetic and numeric characters.

★ Label: My Field 3

This is the name to be shown on the screens where the field is active.

★ Field order: 3

This is the order in which this field will be shown on the screens where is active.

Validity: valid

Field type: Checkbox

Object type: Ticket

Default value: Unchecked

This is the default value for this field.

Рисунок: Окно настройки Динамического Поля Checkbox.

10.2.5. Настройка Динамического Поля Dropdown (Выпадающий Список)

Динамическое поле Выпадающий Список (Dropdown) используется для хранения одного значения, которое выбирается из закрытого списка.

Настройки динамического поля Dropdown:

- Возможные значения: Обязательное.

  List of values to choose. If used, a new value is necessary to specify the Key (internal value) and the Value (display value).

- Значение по умолчанию: Необязательное.

  This is the value to be show by default on the edit screens (like New Phone Ticket or Ticket Compose), the default value for this field is closed selection defined by the Possible values.

- Добавить пустое значени: Обязательное, (Да / Нет).

  Если эта опция включена, дополнительная стоимость определяется для отображения "-" в списке возможных значений, это специально значение пустоне внутри.

- Переводимые значения: Обязательное, (Да / Нет).

  This setting is used mark the possible values of this field to be translated. Only the display values are translated, internal values are not affected, the translation of the values needs to be manually added to the language files.

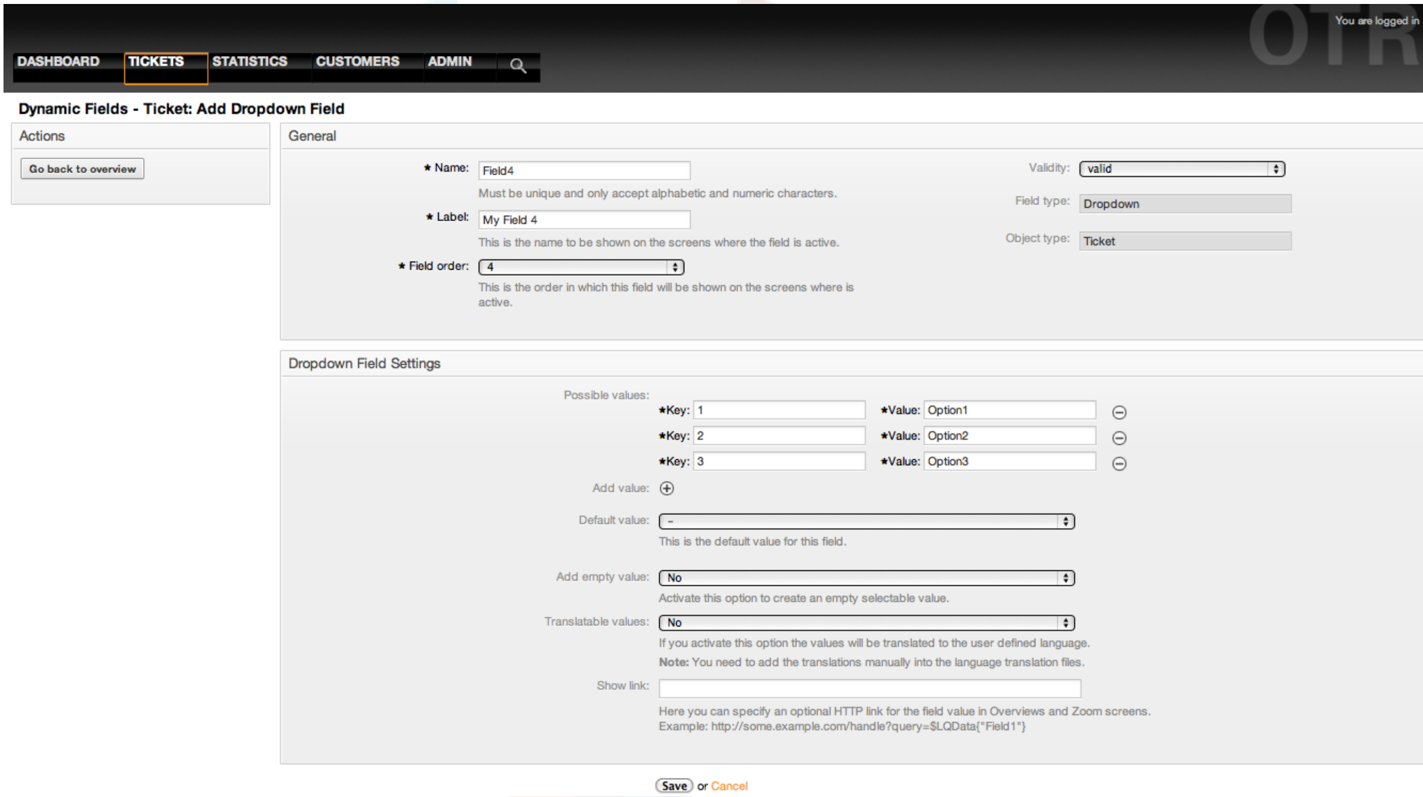
- Показать ссылку: Необязательное.

Если установлено, то значение поля будет установлено в кликабельные ссылки.

Например, если ссылка Показать установлена в значение "http://www.otrs.com", нажатие на значение поля приведет к открытию веб-страницы OTRS.

Примечание

The use of $LQData{"NameX"} in the Set link value, where NameX is the name of the field, will add the field value as part of the link reference.



Dynamic Fields - Ticket: Add Dropdown Field

★ Name: Field4
Must be unique and only accept alphabetic and numeric characters.
★ Label: My Field 4
This is the name to be shown on the screens where the field is active.
★ Field order: 4
This is the order in which this field will be shown on the screens where is active.

Validity: valid
Field type: Dropdown
Object type: Ticket

Possible values:
★Key: 1 ★Value: Option1
★Key: 2 ★Value: Option2
★Key: 3 ★Value: Option3
Add value:
Default value: -
This is the default value for this field.
Add empty value: No
Activate this option to create an empty selectable value.
Translatable values: No
If you activate this option the values will be translated to the user defined language.
Note: You need to add the translations manually into the language translation files.
Show link:
Here you can specify an optional HTTP link for the field value in Overviews and Zoom screens.
Example: http://some.example.com/handle?query=$LQData{"Field1"}

Рисунок: Окно настройки динамического поля Dropdown.

10.2.6. Конфигурация Динамического Поля Multiselect (множественный выбор)

Multiselect dynamic field is used to store multiple values, from a closed list.

Настройка Динамического Поля Multiselect (множественный выбор):

- Возможные значения: Обязательное.

  List of values to choose from. When adding additional list items, it is necessary to specify the Key (internal value) and the Value (display value).

- Значение по умолчанию: Необязательное.

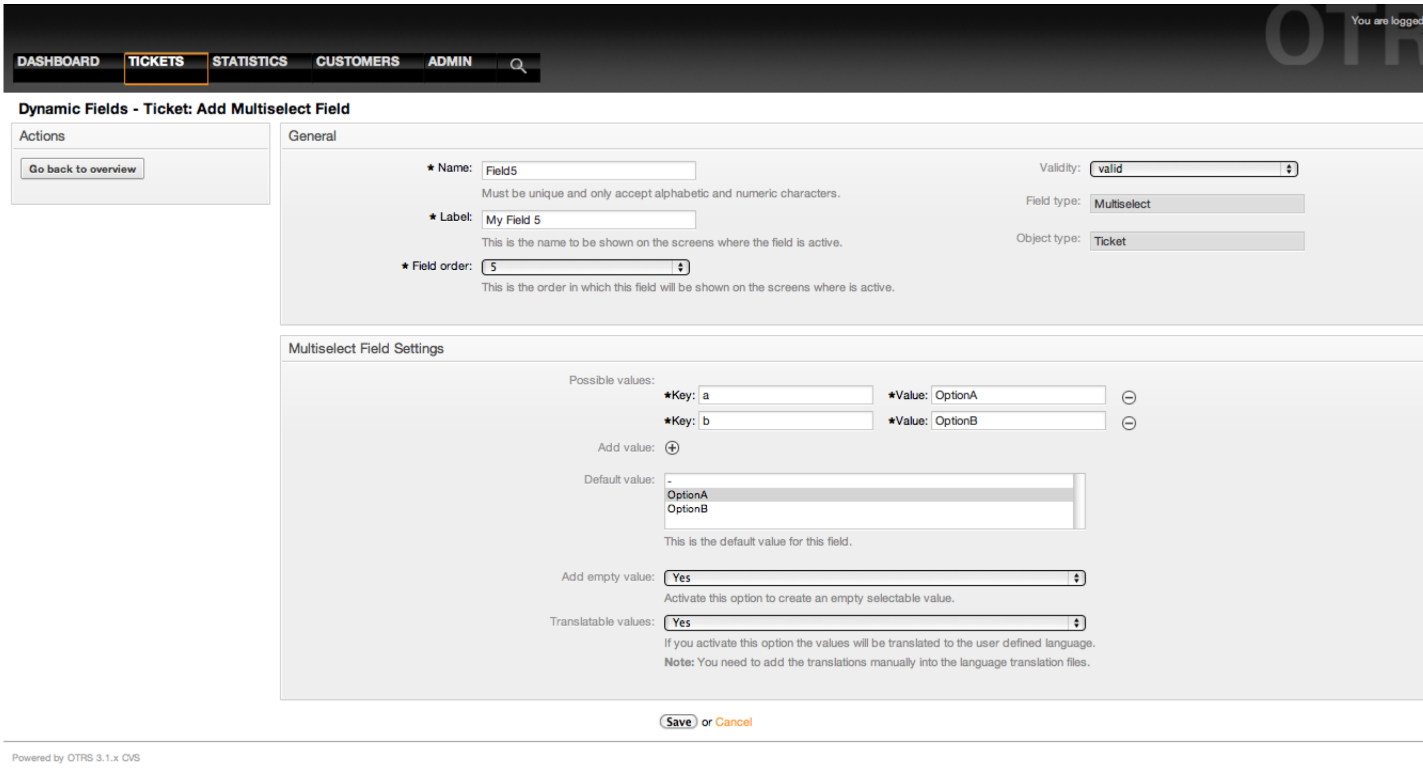
  This is the value to be shown by default on the edit screens (like New Phone Ticket or Ticket Compose), the default value for this field is closed selection as defined by the Possible values.

- Добавить пустое значени: Обязательное, (Да / Нет).

If this option is activated an extra value is defined to show as a "-" in the list of possible values. This special value is empty internally.

- Переводимые значения: Обязательное, (Да / Нет).

  This setting is used mark the possible values of this field to be translated. Only the display values are translated, internal values are not affected, the translation of the values needs to be manually added to the language files.



Dynamic Fields - Ticket: Add Multiselect Field

Actions

General

* Name: Field5

Must be unique and only accept alphabetic and numeric characters.

* Label: My Field 5

This is the name to be shown on the screens where the field is active.

* Field order: 5

This is the order in which this field will be shown on the screens where is active.

Validity: valid

Field type: Multiselect

Object type: Ticket

Multiselect Field Settings

Possible values:

*Key: a *Value: OptionA

*Key: b *Value: OptionB

Add value:

Default value: - / OptionA / OptionB

This is the default value for this field.

Add empty value: Yes

Activate this option to create an empty selectable value.

Translatable values: Yes

If you activate this option the values will be translated to the user defined language.

Note: You need to add the translations manually into the language translation files.

Рисунок: Окно настройки динамического поля Multiselect.

10.2.7. Настройка Динамического Поля Date

Динамическое поле Date используется для хранения значения (День, Месяц и Год).

Настройки динамического поля Date (Дата):

- Разница дат по умолчанию: Опционально, Целое.

  Количество секунд (положительное или отрицательное значение) между текущей датой и выбраной дантой, которое будет отображаться на страницах редактирования по умолчанию (например на странице Новая Заявка на основе телефонного звонка или Создать Заявку).

- Определить период лет: Обязательно (Да / Нет).

  Used to set a defined number of years in the past and the future based on the current date of the year select for this field, If set to Yes the following options are available:

  - Лет в прошлом: Опционально, положительное целое число.

Define the number of years in the past from the current day to display in the year selection for this dynamic field in edit screens.

- Лет в будущем: Опционально, Положительное целое.

  Define the number of years in the future from the current day to display in the year selection for this dynamic field in edit screens.

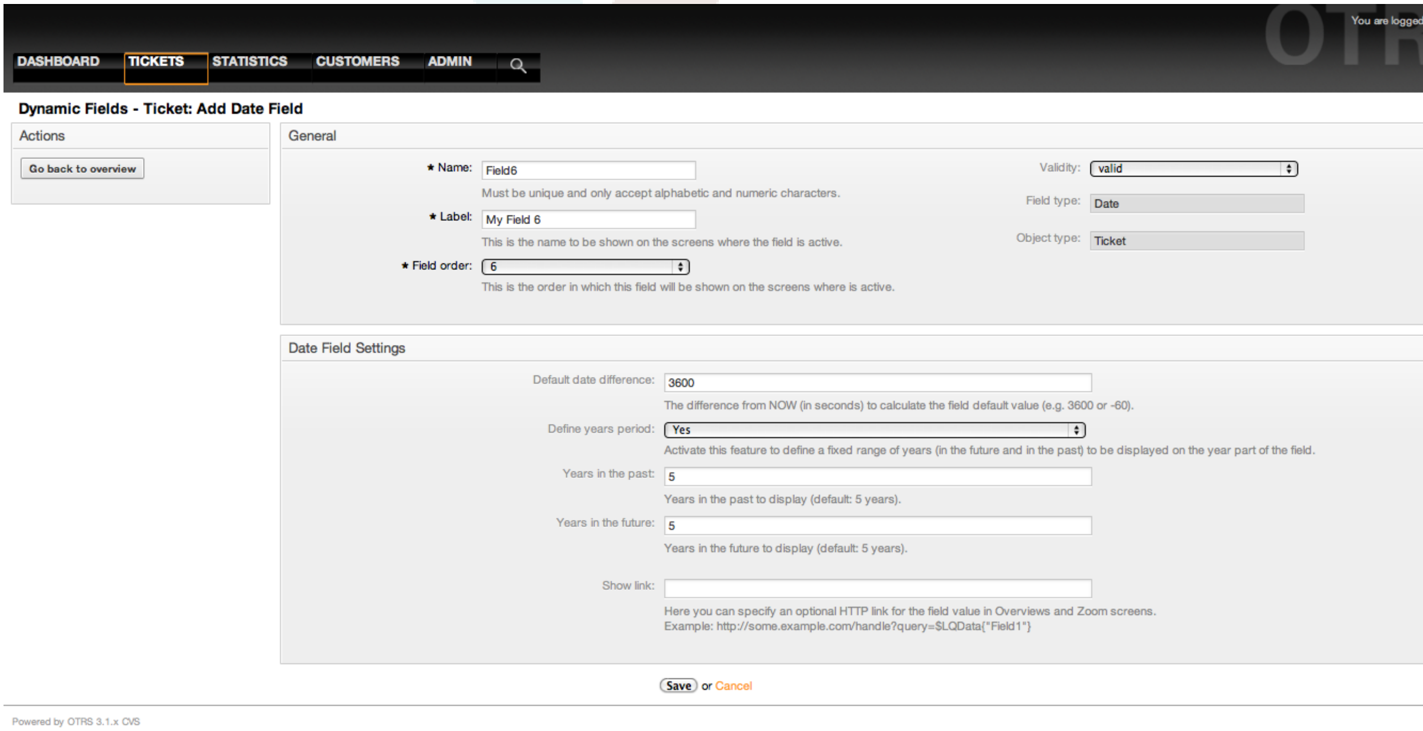
- Показать ссылку: Необязательное.

  Если установлено, то значение поля будет установлено в кликабельные ссылки.

  Например, если ссылка Показать установлена в значение "http://www.otrs.com", нажатие на значение поля приведет к открытию веб-страницы OTRS.

Примечание

The use of $LQData{"NameX"} in the Set link value, where NameX is the name of the field will add the field value as part of the link reference.



Dynamic Fields - Ticket: Add Date Field

General

* Name: Field6
Must be unique and only accept alphabetic and numeric characters.
* Label: My Field 6
This is the name to be shown on the screens where the field is active.
* Field order: 6
This is the order in which this field will be shown on the screens where is active.

Validity: valid
Field type: Date
Object type: Ticket

Date Field Settings

Default date difference: 3600
The difference from NOW (in seconds) to calculate the field default value (e.g. 3600 or -60).
Define years period: Yes
Activate this feature to define a fixed range of years (in the future and in the past) to be displayed on the year part of the field.
Years in the past: 5
Years in the past to display (default: 5 years).
Years in the future: 5
Years in the future to display (default: 5 years).
Show link:
Here you can specify an optional HTTP link for the field value in Overviews and Zoom screens.
Example: http://some.example.com/handle?query=$LQData{"Field1"}

Powered by OTRS 3.1.x CVS

Рисунок: Окно настройки динамического поля Date.

10.2.8. Настройка Динамического Поля Date / Time

Динамическое поле Date / Time используется для хранения значения даты и времени (Минута, Час, День, Месяц и Год).

Настройки динамического поля Date / Time:

- Разница дат по умолчанию: Опционально, Целое.

Number of seconds (positive or negative) between the current date and the selected date to be shown by default in the edit screens (like New Phone Ticket or Ticket Compose).

- Определить период лет: Обязательно (Да / Нет).

  Used to set a defined number of years in the past and the future from thee current date in the year select of this field, If set to Yes the following options are available:

  - Лет в прошлом: Опционально, положительное целое число.

    Define the number of years in the past from the current day to display in the year selection for this dynamic field in edit screens.

  - Лет в будущем: Опционально, Положительное целое.

    Define the number of years in the future from the current day to display in the year selection for this dynamic field in edit screens.

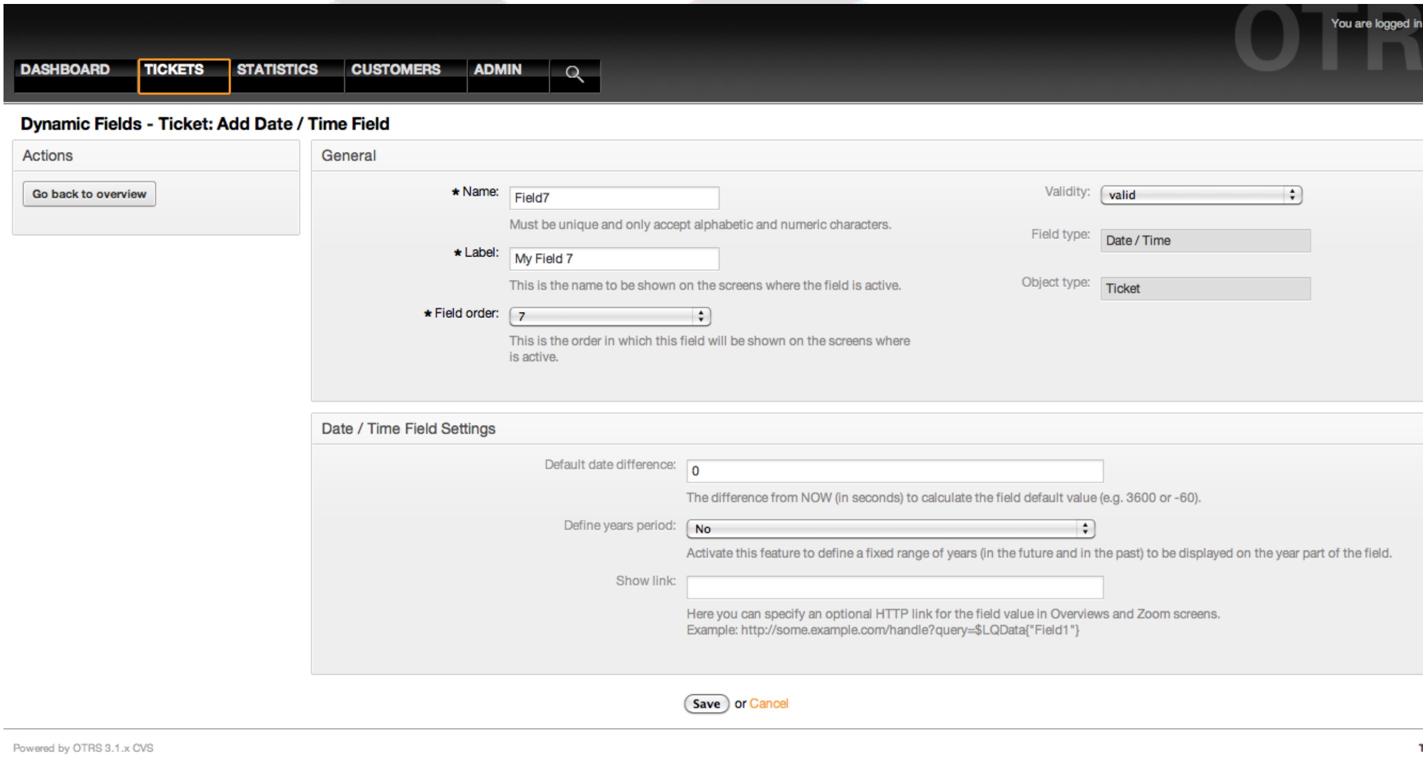
- Показать ссылку: Необязательное.

  Если установлено, то значение поля будет установлено в кликабельные ссылки.

  Например, если ссылка Показать установлена в значение "http://www.otrs.com", нажатие на значение поля приведет к открытию веб-страницы OTRS.

Примечание

The use of $LQData{"NameX"} in the Set link value, where NameX is the name of the field will add the field value as part of the link reference.



Dynamic Fields - Ticket: Add Date / Time Field

General

* Name: Field7
Must be unique and only accept alphabetic and numeric characters.
* Label: My Field 7
This is the name to be shown on the screens where the field is active.
* Field order: 7
This is the order in which this field will be shown on the screens where is active.

Validity: valid
Field type: Date / Time
Object type: Ticket

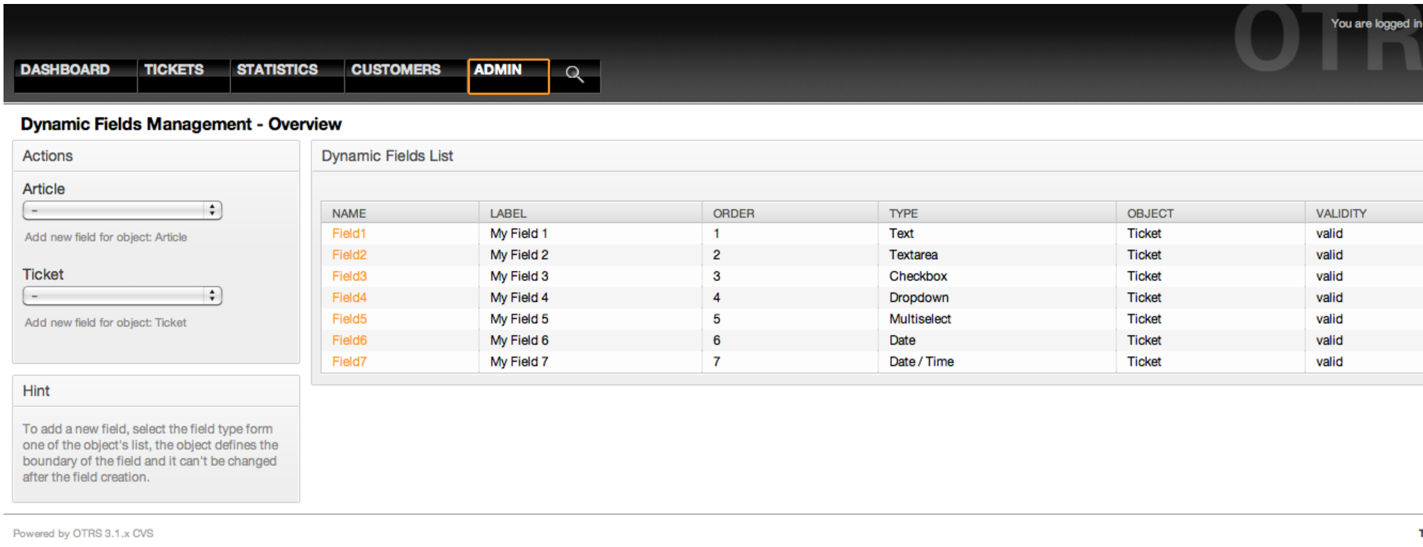
Date / Time Field Settings

Default date difference: 0
The difference from NOW (in seconds) to calculate the field default value (e.g. 3600 or -60).
Define years period: No
Activate this feature to define a fixed range of years (in the future and in the past) to be displayed on the year part of the field.
Show link:
Here you can specify an optional HTTP link for the field value in Overviews and Zoom screens.
Example: http://some.example.com/handle?query=$LQData{"Field1"}

Рисунок: Окно настройки динамического поля Date / Time.

10.2.9. Редактирование Динамического Поля

Страница просмотра заполненого динамического поля (с предыдущими примерами) должна выглядеть так:



Dynamic Fields Management - Overview

Actions

Article

Add new field for object: Article

Ticket

Add new field for object: Ticket

Hint

To add a new field, select the field type form one of the object's list, the object defines the boundary of the field and it can't be changed after the field creation.

Dynamic Fields List

NAME	LABEL	ORDER	TYPE	OBJECT	VALIDITY
Field1	My Field 1	1	Text	Ticket	valid
Field2	My Field 2	2	Textarea	Ticket	valid
Field3	My Field 3	3	Checkbox	Ticket	valid
Field4	My Field 4	4	Dropdown	Ticket	valid
Field5	My Field 5	5	Multiselect	Ticket	valid
Field6	My Field 6	6	Date	Ticket	valid
Field7	My Field 7	7	Date / Time	Ticket	valid

Powered by OTRS 3.1.x CVS

Рисунок: Окно просмотра Динамического поля заполненного тестовыми данными.

To change or edit a dynamic field you must have at least one field defined, select an already added field from the dynamic fields overview screen and update its settings.

Примечание

Не все настройки динамического поля могут быть изменены, тип Поля и тип Объекта не могут быть изменены.

It is not recommended to change the field internal name, but the label can be changed at any time. If internal name is changed all "SysConfig" settings that have a reference to that particular field needs to be updated as well as user preferences (if defined).

10.2.10. Отображение Динамического Поля на Экране

Есть два обязательных условия, которые необходимо выполнить, чтобы отобразить динамическое поле на определенных страницах:

1. Динамическое поле должно быть действительным

2. Динамическое поле должно быть установлено в 1 или 2 на странице с настройками.

Follow these steps to show a dynamic field in a screen

- Убедитесь что значение динамического поля установлено правильно. Убедится в правильности значения можно посмотрев на страницу просмотра. Немедленно отредактируйте значение, чтобы сделать его правильным.

- Open the "sysconfig" and select "Ticket" from the dropdown list in the Actions side bar located in the left part of the screen.

  ### Примечание

  You can also search for "DynamicField" in the search box above or the "sysconfig" key directly if you already know it.

- Locate the setting sub-group for the screen that you are looking for and click on it. For example "Frontend::Agent::Ticket::ViewPhoneNew".

- Search for the setting that ends with "###DynamicField". For example "Ticket::Frontend::AgentTicketPhone###DynamicField".

- If the setting is empty or does not have the required dynamic field name, click on the "+" button to add a new entry. For example Key: Field1, Content: 1.

  If the setting already has the dynamic field name listed be sure that is set to "1" to display the field or to "2" to display it as mandatory.

- Save the configuration by clicking in the "Update" button and the bottom of the screen and navigate to the screen where you want the field to be displayed.

10.2.10.1. Показать Примеры

The following are "sysconfig" configurations examples to show or hide dynamic fields on different screens.

Пример 4.21. Активировать Field1 на странице Новая Заявка на основе телефонного звонка.

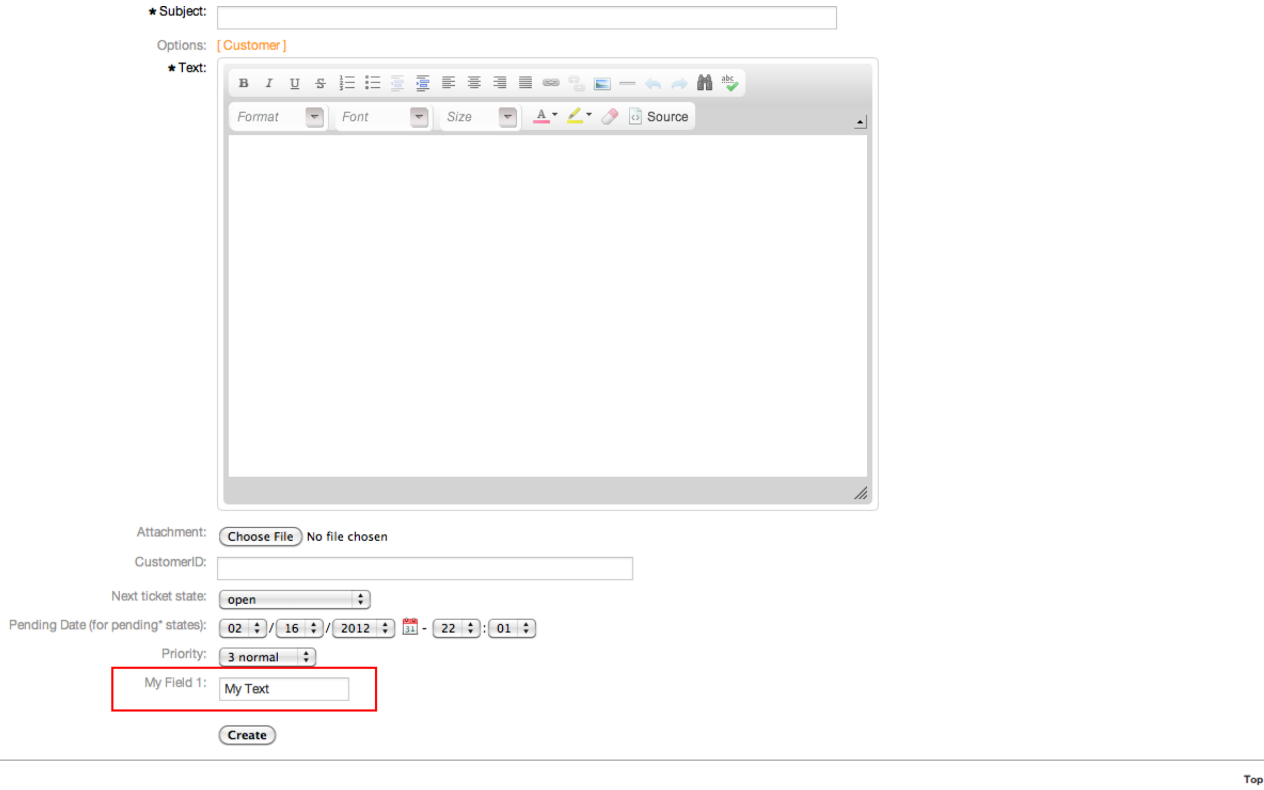
- *Групировать:* Заявку

- *Подгруппа:* Frontend::Agent::Ticket::ViewPhoneNew

- *Настройка:* Ticket::Frontend::AgentTicketPhone###DynamicField

- *Значение:*

Ключ	Content
Поле1	1



★ Subject:

Options: [Customer]

★ Text:

Attachment: Choose File No file chosen

CustomerID:

Next ticket state: open

Pending Date (for pending* states): 02 / 16 / 2012 - 22 : 01

Priority: 3 normal

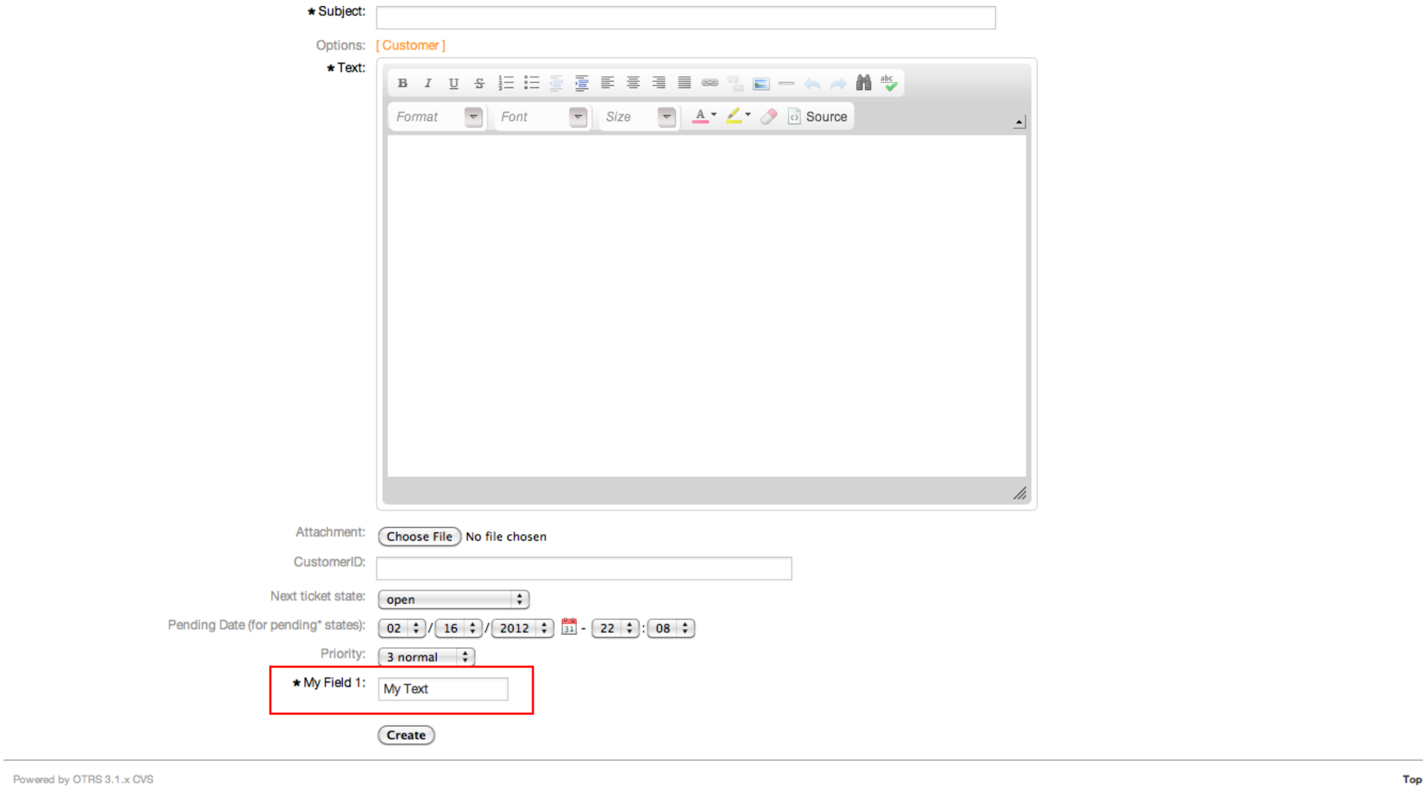
My Field 1: My Text

Рисунок:. Экран приветствия.

Пример 4.22. Сделать Field1 на странице просмотра Заявки созданной на базе телефонного звонка New Phone Ticket Screen as mandatory.

- *Групировать:* Заявку
- *Подгруппа:* Frontend::Agent::Ticket::ViewPhoneNew
- *Настройка:* Ticket::Frontend::AgentTicketPhone###DynamicField
- *Значение:*

Ключ	Content
Поле1	2



⋆ Subject:

Options: [Customer]

⋆ Text:

Attachment: Choose File No file chosen

CustomerID:

Next ticket state: open

Pending Date (for pending* states): 02 / 16 / 2012 - 22 : 08

Priority: 3 normal

⋆ My Field 1: My Text

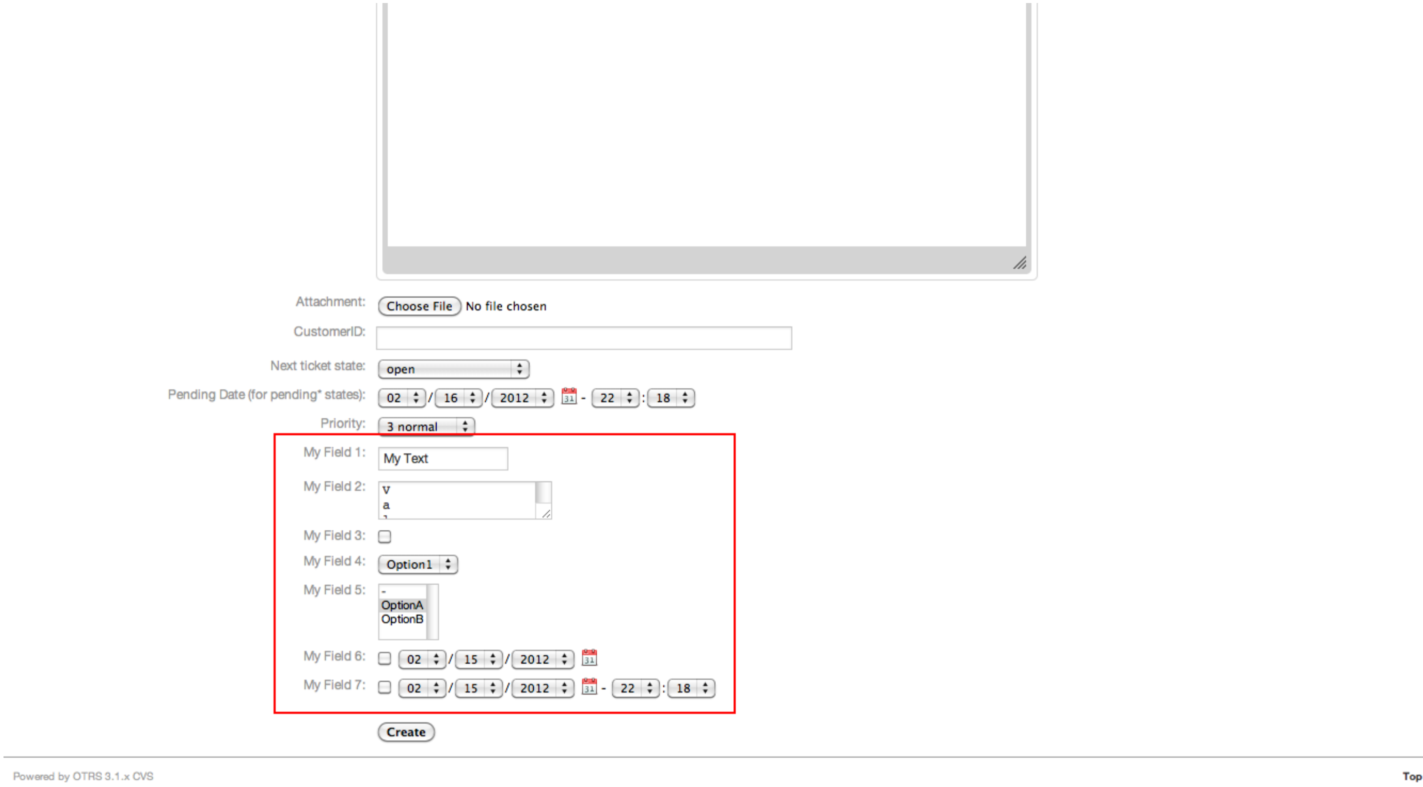
Create

Рисунок: поле Field1 на странице Новая Заявка на основе телефонного звонка объязательное.

Пример 4.23. Активировать несколько полей на странице просмомра Новых Телефонных Заявко.

- *Групировать:* Заявку
- *Подгруппа:* Frontend::Agent::Ticket::ViewPhoneNew
- *Настройка:* Ticket::Frontend::AgentTicketPhone###DynamicField
- *Значение:*

Ключ	**Content**
Поле1	1
Поле2	1
Поле3	1
Поле4	1
Поле5	1
Field6	1
Поле7	1



Attachment: Choose File No file chosen

CustomerID:

Next ticket state: open

Pending Date (for pending* states): 02 / 16 / 2012 - 22 : 18

Priority: 3 normal

My Field 1: My Text

My Field 2: v a

My Field 3: □

My Field 4: Option1

My Field 5: - OptionA OptionB

My Field 6: □ 02 / 15 / 2012

My Field 7: □ 02 / 15 / 2012 - 22 : 18

Create

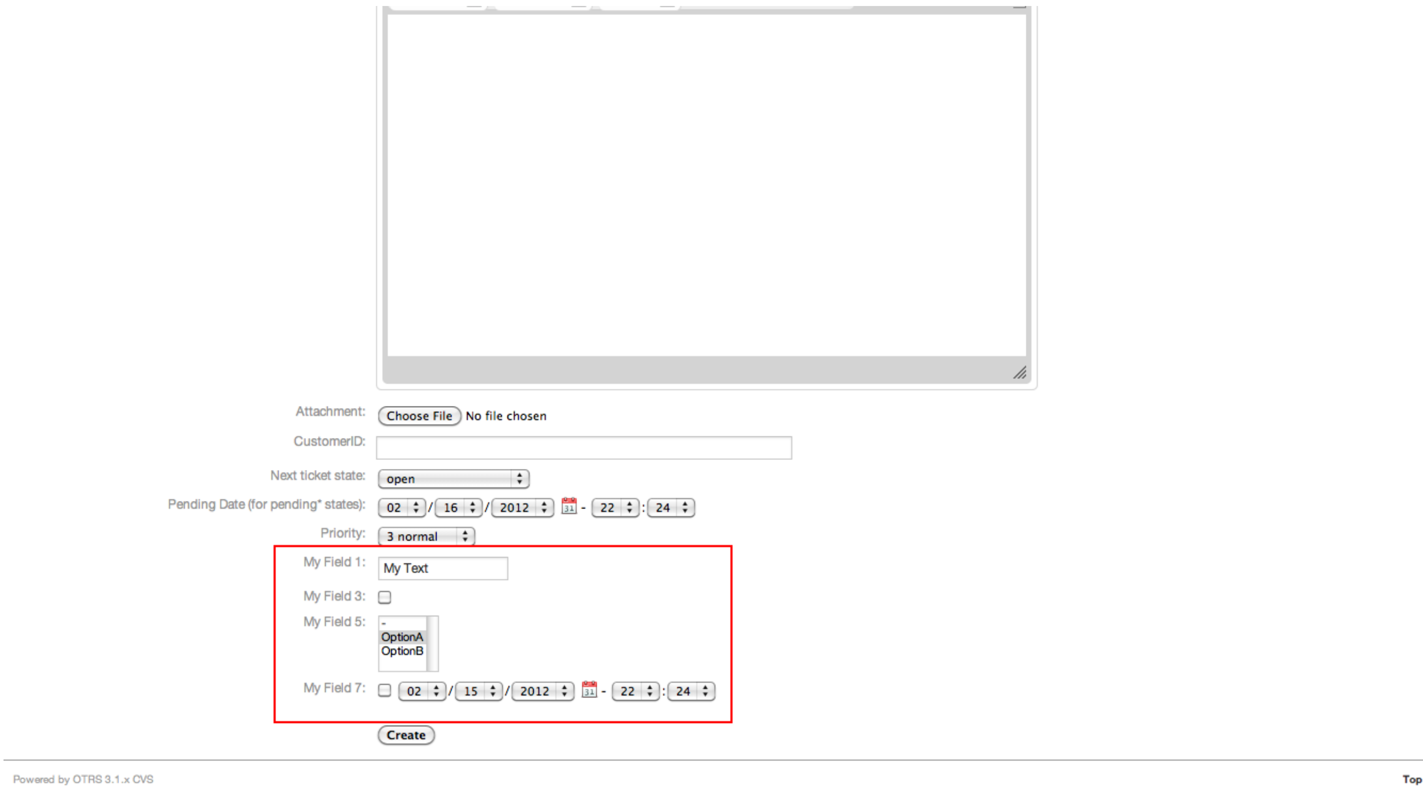
Powered by OTRS 3.1.x CVS

Рисунок: Несколько полей на странице Заявки созданной на основе телефонного звонка обязательны

Пример 4.24. Отключить некоторые поля на странице Новой Заявки, созданной на основе телефонного звонка.

- *Групировать:* Заявку
- *Подгруппа:* Frontend::Agent::Ticket::ViewPhoneNew
- *Настройка:* Ticket::Frontend::AgentTicketPhone###DynamicField
- *Значение:*

Ключ	**Content**
Поле1	1
Поле2	0
Поле3	1
Поле4	0
Поле5	1
Field6	0
Поле7	1



Attachment: Choose File No file chosen

CustomerID:

Next ticket state: open

Pending Date (for pending* states): 02 / 16 / 2012 - 22 : 24

Priority: 3 normal

My Field 1: My Text

My Field 3: □

My Field 5: - OptionA OptionB

My Field 7: □ 02 / 15 / 2012 - 22 : 24

Create

Powered by OTRS 3.1.x CVS

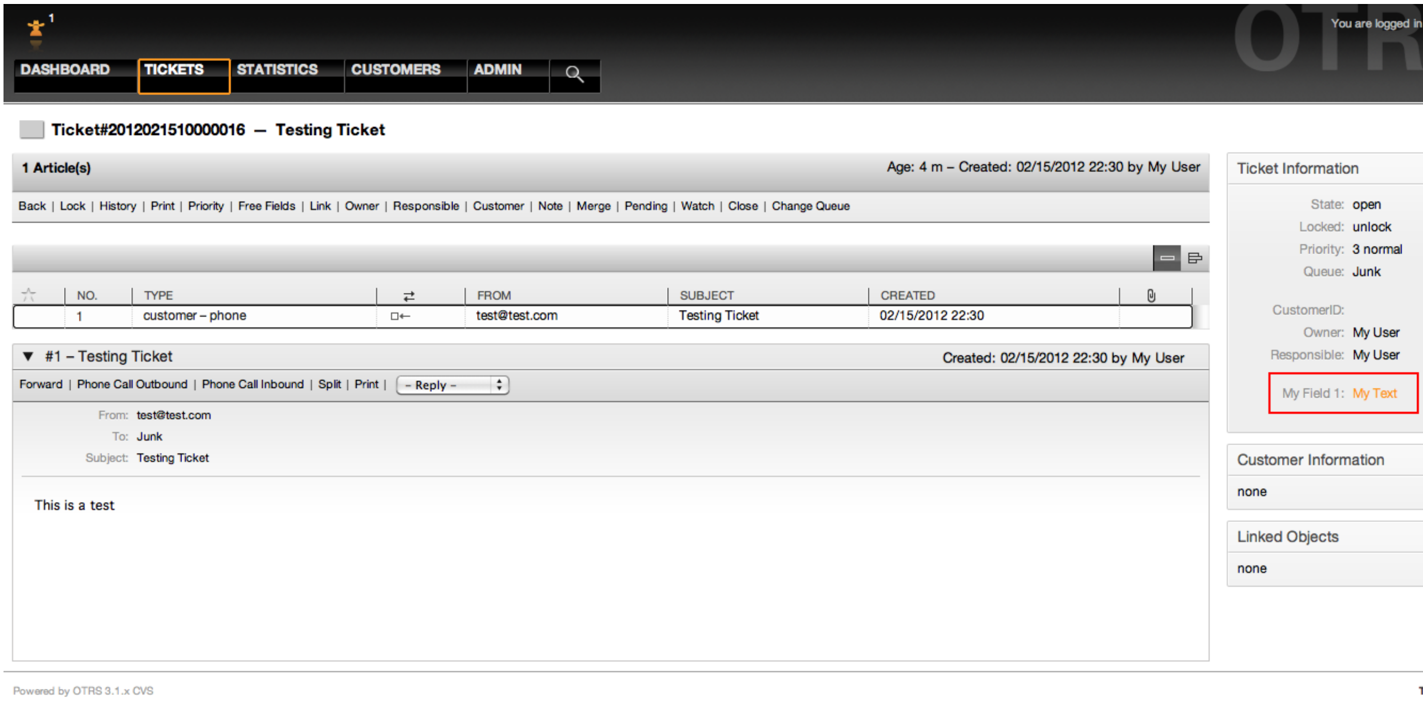
Top

Figure: Some deactivated fields in New Phone Ticket Screen as mandatory.

Пример 4.25. Активировать поле Field1 на странице просмотра подробной информации о Заявке.

- *Групировать:* Заявку
- *Подгруппа:* Frontend::Agent::Ticket::ViewZoom
- *Установка:* Ticket::Frontend::AgentTicketZoom###DynamicField
- *Значение:*

Ключ	**Content**
Поле1	1


DASHBOARD
TICKETS
STATISTICS
CUSTOMERS
ADMIN
You are logged in
Back | Lock | History | Print | Priority | Free Fields | Link | Owner | Responsible | Customer | Note | Merge | Pending | Watch | Close | Change Queue
Forward | Phone Call Outbound | Phone Call Inbound | Split | Print | – Reply –
Ticket Information
Customer Information
Linked Objects
Powered by OTRS 3.1.x CVS

Ticket#2012021510000016 — Testing Ticket

1 Article(s)

Age: 4 m – Created: 02/15/2012 22:30 by My User

NO.	TYPE	⇄	FROM	SUBJECT	CREATED
1	customer – phone	□←	test@test.com	Testing Ticket	02/15/2012 22:30

▼ #1 – Testing Ticket

Created: 02/15/2012 22:30 by My User

From: test@test.com
To: Junk
Subject: Testing Ticket

This is a test

State: open
Locked: unlock
Priority: 3 normal
Queue: Junk
CustomerID:
Owner: My User
Responsible: My User
My Field 1: My Text

none

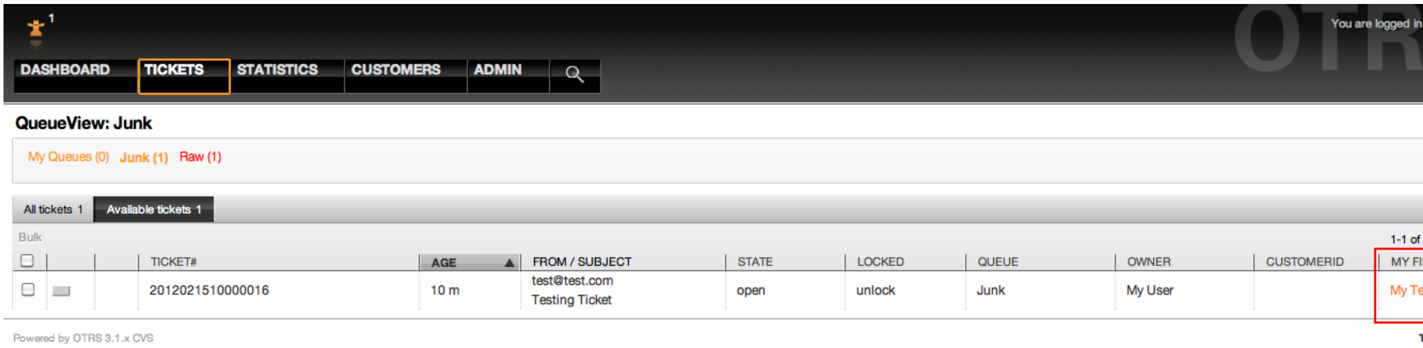
none

Figure: Field1 in Ticket Zoom Screen.

Пример 4.26. Активировать поле Field1 на странице просмотра обобщенной информации о Заявке.

- *Групировать:* Заявку
- *Подгруппа:* Frontend::Agent::TicketOverview
- *Настройка:* Ticket::Frontend::OverviewSmall###DynamicField
- *Значение:*

Ключ	Content
Поле1	1



QueueView: Junk

My Queues (0) Junk (1) Raw (1)

All tickets 1 | Available tickets 1

Bulk

1-1 of

		TICKET#	AGE	FROM / SUBJECT	STATE	LOCKED	QUEUE	OWNER	CUSTOMERID	MY FI
□		2012021510000016	10 m	test@test.com Testing Ticket	open	unlock	Junk	My User		My Te

Powered by OTRS 3.1.x CVS

Figure: Field1 in Ticket Overview Small Screen.

Этот параметр влияет на шаблоны: Эскалации, Заблокированные, Очереди, Ответственные, Статус и Watch-шаблон.

10.2.11. Установка Значения по Умолчанию с помощью модуля Ticket Event

Событие Заявки (например TicketCreate) может вызывать множество значений для определенного поля, если его значение еще до сих пор не установлено.

Примечание

By using this method this default value, is not seen in the edit screen (e.g. New Phone Ticket) since the value is set after the creation of the ticket.

To activate this feature it is necessary to enable the following setting: "Ticket::EventModulePost###TicketDynamicFieldDefault".

Пример 4.27. Активировать Field1 в событии TicketCreate.

- *Групировать:* Заявку
- *Подгруппа:* Core::TicketDynamicFieldDefault
- *Настройка:* Ticket::TicketDynamicFieldDefault###Element1

Примечание

Эта настройка может быть установлена в любую из 16 Ticket::TicketDynamicFieldDefault###Element.

If more that 16 fields needs to be set up a custom XML file must be places in $OTRS_HOME/Kernel/Config/files directory to extend this feature.

- *Значение:*

Ключ	**Content**
Событие	TicketCreate
Название	Поле1
Значение	новое значение

10.2.12. Установить значение по умолчанию для Пользовательских Настроек

Значение по умолчанию для динамического поля может быть перезаписано значением, определенным пользователем, которое хранится в пользовательских настройках.

Using this method, the default value of the field will be shown on any screen where the field is activated (if the field does not have already a different value).

The "sysconfig" setting "PreferencesGroups###DynamicField" located in the "Frontend::Agent::Preferences" Sub-group. This setting is an example of how to create an entry in the User Preferences screen to set an exclusive dynamic field default value for the selected user. The limitation of this setting is that it only permits the use of one dynamic field. If two or more fields will use this feature, it is necessary to create a custom XML configuration file to add more settings similar to this one.

Примечание

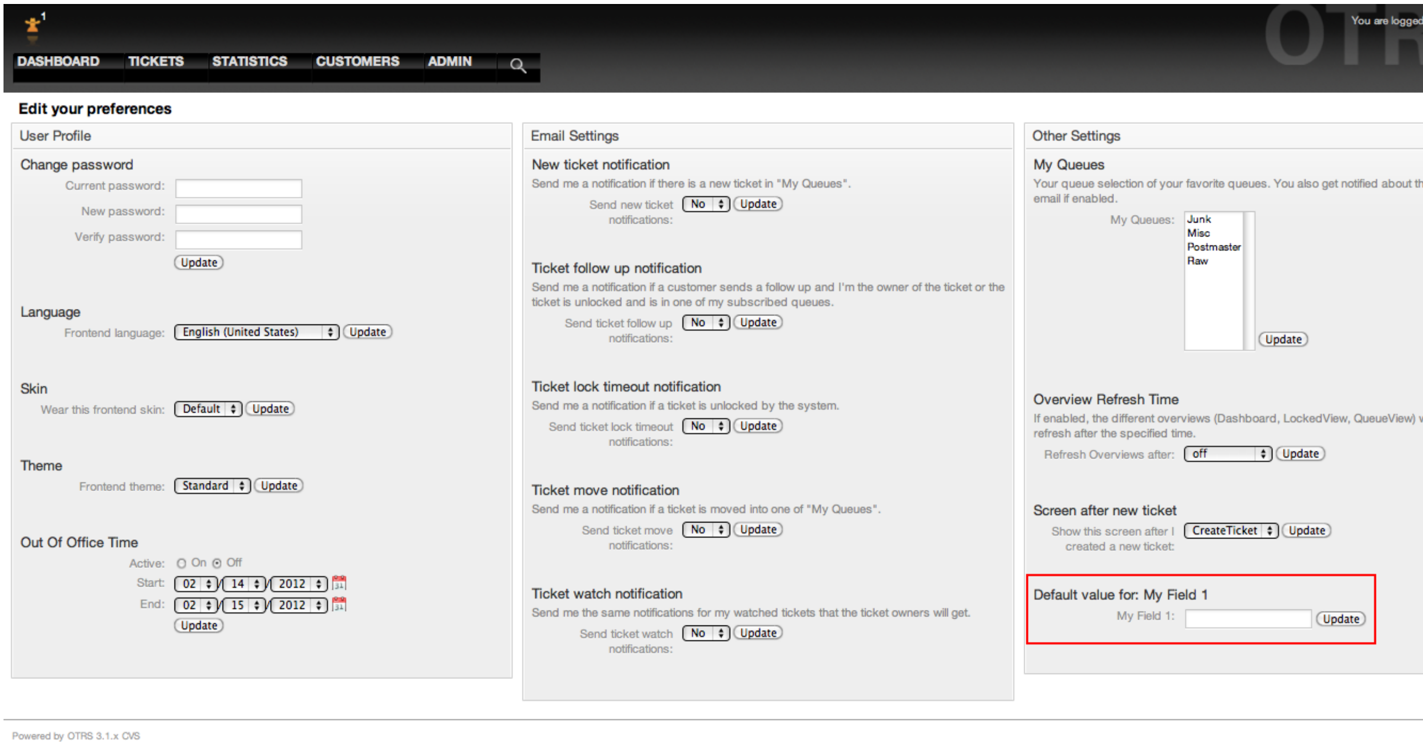
Remember, if more settings are added in a new XML each setting name needs to be unique in the system and different than "PreferencesGroups###DynamicField". for example: PreferencesGroups###101-DynamicField-Field1, PreferencesGroups###102-DynamicField-Field2, PreferencesGroups###My-Field1, PreferencesGroups###My-Field2, etc.

Пример 4.28. Активировать поле Field1 на странице Пользовательские Настройки.

- *Групировать:* Заявку
- *Подгрупа:* Frontend::Agent::Preferences
- *Настройка:* PreferencesGroups###101-DynamicField-Field1
- *Значение:*

Ключ	**Content**
Событие	TicketCreate
Активный	1
Block	Input
Поле	Другие настройки
Data:	$Env{"UserDynamicField_Field1"}
Ключ:	Мое Поле 1

Ключ	Content
Метка:	Значение по умолчанию для: My Field 1
Module:	Kernel::Output::HTML::PreferencesGeneric
PrefKey:	UserDynamicField_Field1
Prio:	7000



Edit your preferences

User Profile

Change password
Current password:
New password:
Verify password:
Update

Language
Frontend language: English (United States) Update

Skin
Wear this frontend skin: Default Update

Theme
Frontend theme: Standard Update

Out Of Office Time
Active: ○ On ● Off
Start: 02 / 14 / 2012
End: 02 / 15 / 2012
Update

Email Settings

New ticket notification
Send me a notification if there is a new ticket in "My Queues".
Send new ticket notifications: No Update

Ticket follow up notification
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.
Send ticket follow up notifications: No Update

Ticket lock timeout notification
Send me a notification if a ticket is unlocked by the system.
Send ticket lock timeout notifications: No Update

Ticket move notification
Send me a notification if a ticket is moved into one of "My Queues".
Send ticket move notifications: No Update

Ticket watch notification
Send me the same notifications for my watched tickets that the ticket owners will get.
Send ticket watch notifications: No Update

Other Settings

My Queues
Your queue selection of your favorite queues. You also get notified about th email if enabled.
My Queues: Junk
Misc
Postmaster
Raw
Update

Overview Refresh Time
If enabled, the different overviews (Dashboard, LockedView, QueueView) w refresh after the specified time.
Refresh Overviews after: off Update

Screen after new ticket
Show this screen after I created a new ticket: CreateTicket Update

Default value for: My Field 1
My Field 1:
Update

Powered by OTRS 3.1.x CVS

Рисунок: Field1 в окне с пользовательских настроек агентов.

11. Generic Interface

The OTRS Generic Interface consists of a multiple layer framework that lets OTRS communicate with other systems via a web service. This communication could be bi-directional:

- *OTRS as Provider:* OTRS acts as a server listening to requests from the External System, processing the information, performing the requested action, and answering the request.
- *OTRS as Requester:* OTRS acts as a client collecting information, sending the request to the Remote System, and waiting for the response.

11.1. Слои интерфейса Generic Interface

Generic Interface is build based on a layer model, to be flexible and easy to customize.

A layer is a set of files, which control how the Generic Interface performs different parts of a web service. Using the right configuration, one can build different web services for different External Systems without creating new modules.

Примечание

If the Remote System does not support the current bundled modules of the Generic Interface, special modules need to be developed for that specific web service.

Перечень модулей предоставляемых Generic Interface поставляется вместе с OTRS и будет обновлен или увеличен с течением времени.

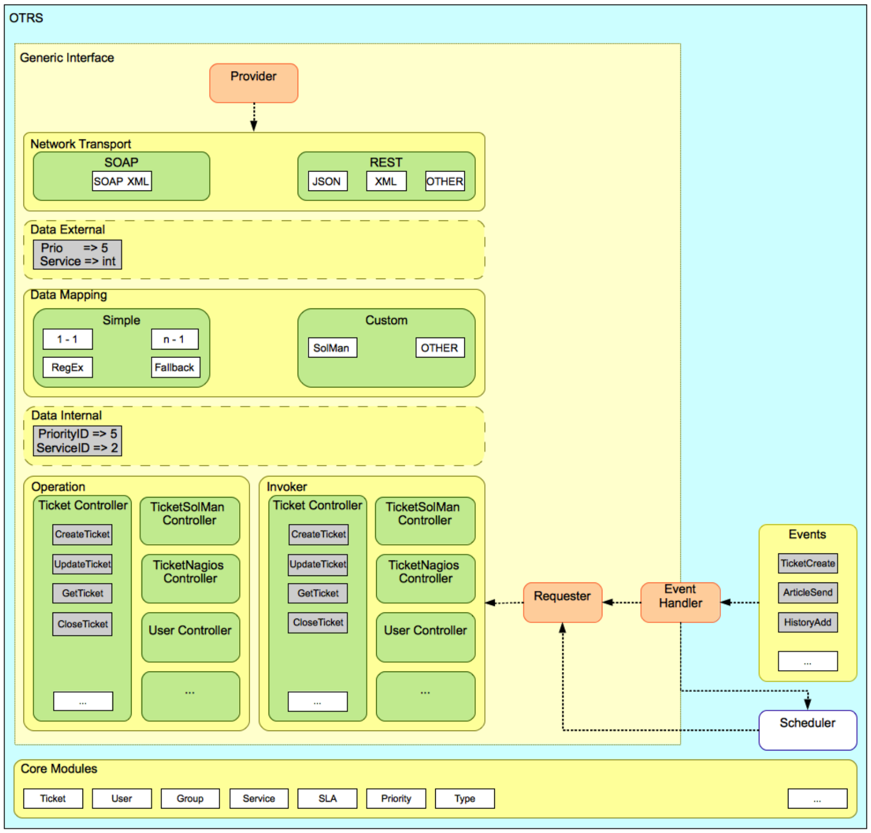


Рисунок: Слои графического интерфейса

11.1.1. Транспортная Сеть

This layer is responsible for the correct communication with the Remote System. It receives requests and generates responses when acting as provider, and generates requests and receives responses when acting as requester.

Provider communication is handled by a new web server handle called "nph-genericinterface.pl".

Requester communication could be initiated during an event triggered by a Generic Interface module or any other OTRS module. This event is catched by the event handler and depending on the configuration the event will be processed directly by the requester object or delegated to the Scheduler (a separated deamon designed to process tasks asynchronously).

11.1.2. Отображение Данных

This layer is responsible for translating data structures between OTRS and the Remote System (data internal and data external layers). Usually Remote Systems have different data structures than OTRS (including different values and names for those values), and here resides the importance of the layer to change the received information into something that OTRS can understand and on the opposite way send the information to each Remote System using their data dictionaries.

Example: "Priority" (OTRS) might be called "Prio" in a remote system and it could be that value "1 Low" (OTRS) should be mapped to "Information" on the remote system.

11.1.3. Контроллер

Controllers are collections of similar Operations or Invokers. For example, a Ticket controller might contain several standard ticket operations. Custom controllers can be implemented, for example a "TicketExternalCompany" controller which may contain similar functions as the standard Ticket controller, but with a different data interface, or function names (to adapt to the Remote System function names) or complete different code.

One application for Generic Interface could be to synchronize information with one Remote System that only can talk with another Remote System of the same kind. In this case new controllers needs to be developed and the Operations and Invokers has to emulate the Remote System behavior in such way that the interface that OTRS exposes is similar to the Remote System's interface.

11.1.4. Операция (OTRS как поставщик услуг)

An Operation is a single action that can be performed within OTRS. All operations have the same programming interface, they receive the data into one specific parameter, and return a data structure with a success status, potential error message and returning data.

Normally operations uses the already mapped data (internal) to call core modules and perform actions in OTRS like: Create a Ticket, Update a User, Invalidate a Queue, Send a Notification, etc. An operation has full access to the OTRS API to perform the action.

11.1.5. Invoker - запрашивающая сторона (OTRS как запрашивающая сторона, клинет)

An Invoker is an action that OTRS performs against a Remote System. Invokers use the OTRS Core modules to process and collect the needed information to create the request. When the information is ready it has to be mapped to the Remote System format in order to be sent to the Remote System, that will process the information execute the action and send the response back, to either process the success or handle errors.

11.2. Поток связи Generic Interface

The Generic Interface has a defined flow to perform actions as a provider and as a requester.

These flows are described below:

11.2.1. OTRS как Поставщик

11.2.1.1. Удаленный Запрос:

1. HTTP-запрос
   - OTRS receives HTTP request and passes it through the layers.
   - The provider module is in charge to execute and control these actions.
2. Транспортная Сеть
   - The network transport module decodes the data payload and separates the operation name from the rest of the data.

- Название операции и данные операции будут возвращены поставщику.

3. *Внешние данные*

   - Данные отправленные из удаленной системы (Это не слой базирующийся на модуле).

4. Mapping

   - Данные преобразуются из формата Внешней Системы во внутренний формат данных OTRS так, как это указано в конфигурации отображения для этой операции (Mapping для входящих данных запроса).

   - Преобразованные данные возвращаются обратно поставщику.

5. *Внутренние Данные*

   - Data as transformed and prepared to be passed to the operation (This is not a module based layer).

6. Операция

   - Принимает и проверяет данные.

   - Осуществляет контроль доступа пользователей.

   - Выполняет действие.

11.2.1.2. OTRS Ответ:

1. Операция

   - Возвращает результирующие данные поставщику.

2. *Внутренние Данные*

   - Данные возвращаемые из операции.

3. Mapping

   - The data is transformed back to the Remote system format as specified in the mapping configuration (Mapping for outgoing response data).

   - Преобразованные данные возвращаются обратно поставщику.

4. *Внешние данные*

   - Data as transformed and prepared to be passed to Network Transport as response.

5. Транспортная Сеть

   - Получает данные уже в формате Удаленной Системы.

   - Создает правильный ответ для этого типа транспортной сети.

6. HTTP ответ

   - Ответ отправляется обратно клиенту веб-сервиса.

- В случае ошибки, сообщение об ошибке отправляется удаденной системе (например SOAP-ошибка, HTTP-ошибка и т.д.).

11.2.2. OTRS как запрашиваемая сторона

11.2.2.1. OTRS Запрос:

1. Обработчки Event Trigger
   - В зависимости от настройки веб-сервис определяет будет ли запрос синхронным или асинхронным.
     - Синхронный
       - A direct call to the Requester is made in order to create a new request and to pass it through the layers.
     - Асинхронный
       - Create a new Generic Interface (Requester) task for the OTRS Scheduler (by delegating the request execution to the Scheduler, the user experience could be highly improved, otherwise all the time needed to prepare the request and the remote execution will be added to the OTRS Events that trigger those requests).
       - In its next cycle the Scheduler process reads the new task and creates a call to the Requester that will create a new request and then passes it through the layers.
2. Запрашивающая сторона
   - Получает данные от событий.
   - Проверяет полученные данные (при необходимости).
   - Вызвать модули ядра для дополнения данных (при необходимости).
   - Возвращает структуру данных запроса или отправляет сигнал "Остановка связи" для Запрашивающей Стороны (requester), чтобы корректно отменить запрос.
3. *Внутренние Данные*
   - Данные передаются от запрашивающей стороны (Это не слой, основанный на модуле).
4. Mapping
   - The data is transformed to the Remote system format as specified in the mapping configuration (Mapping for outgoing response data).
   - Уже преобразованные данные возвращаются Запрашивающей Стороне.
5. *Внешние данные*
   - Данные преобразованы и подготовлены для отправки удаленной системе.
6. Транспортная Сеть
   - Receives the remote operation name and the data already transformed to the Remote System format from the requester.

- Создает действительный запрос для транспортной сети.
- Отправляет запрос удаленной системе и ожидает ответ от нее

11.2.2.2. Удаленный Ответ

1. Транспортная сеть
   - Receives the response and decodes the data payload.
   - Возвращает данные запрашивающей стороне.
2. *Внешние данные*
   - Данные полученные от Удаленной Системы
3. Mapping
   - The data is transformed from the External System format to the OTRS internal format as specified in the mapping configuration for this operation (Mapping for incoming response data).
   - Уже преобразованные данные возвращаются Запрашивающей Стороне.
4. *Внутренние Данные*
   - Data as transformed and ready to be passed back to the requester.
5. Запрашивающая сторона
   - Получает возвращаемые данные.
   - Handles the data as needed specifically by each Invoker (included error handling if any).
   - Return the Invoker result and data to the Requester.
6. Обработчик или Планировщик Действий
   - Receives the data from the Requester, in the case of the Scheduler this data might contain information to Re-Schedule the task immediately or in the future.

11.3. Веб-сервисы

A Web Service is a communication method between two systems, in our case OTRS and a Remote System.

The heart of the Web Service is its configuration, where it is defined what actions the web service can perform internally (Operation), what actions the OTRS request can perform Remote System (Invokers), how data is converted from one system to the other (Mapping), and over which protocol the communication will take place (Transport)

The Generic Interface is the framework that makes it possible to create Web Services for OTRS in a pre-defined way, using already made building blocks that are independent from each other and interchangeable.

11.4. Графический интерфейс веб-сервиса

The web service graphical user interface (GUI) is a tool that allows to construct complex web service configurations in a user friendly and convenient interface. It allows to:

- Создание и Удалинеи веб-сервисов

- Импортировать и Экспортировать настройки (в формате YAML) для существующих веб-сервисов.

- Просматривать, Возвращаться и Экспортировать устаревшие конфигурационные параметры для существующих веб-сервисов на странице истории конфигурации.

- Отслеживать все сообщения журнала для каждого из веб-сервисов на странице Отладчика.

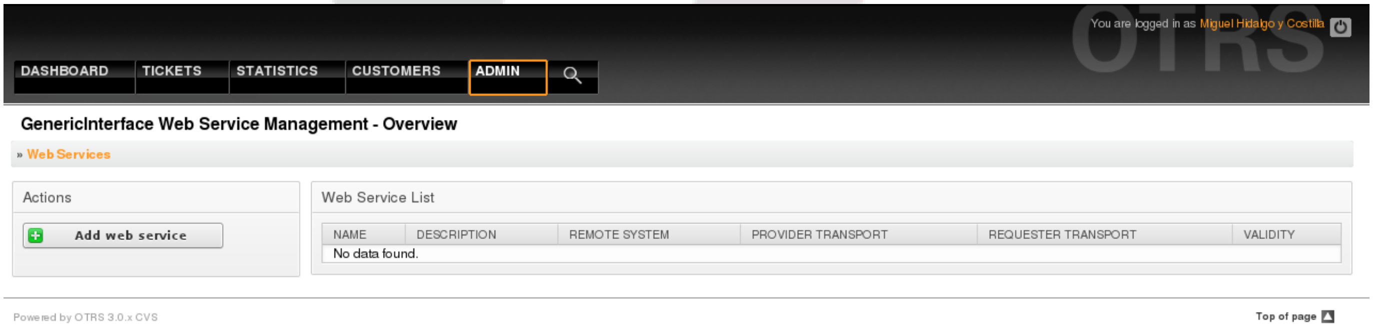
11.4.1. Обзор Веб-интерфейса

The "Web Services" link in the main screen of Admin Interface (in the System Administration box) leads to the web services overview screen, where you are able to manage your web service configurations. You can add new web services or change the configuration of the existing ones from this screen.

Every web service configuration screen has in the upper part of the screen a "bread crumbs" style navigation path. This navigation path is useful to know exactly in which part of the web service configuration we are, and also enables the user to jump back to any part of the configuration process at any time (this action will not save any changes).

Примечание

To create a new web service, press the button "Add web service", and provide the required information.



GenericInterface Web Service Management - Overview

» Web Services

Actions

Web Service List

NAME	DESCRIPTION	REMOTE SYSTEM	PROVIDER TRANSPORT	REQUESTER TRANSPORT	VALIDITY
No data found.					

Powered by OTRS 3.0.x CVS

Рисунок: Обзор Веб-сервисов

11.4.2. Добавить Веб Сервис

The only required field in this part is the web service "Name" that needs to be unique in the system and can not be left empty. Other fields are also necessary for the configuration like the "Debug Threshold" and "Validity" but these fields are already populated with the default value for each list.

The default value for "Debug Threshold" is "debug". When configured in this manner all communication logs are registered in the database. Each subsequent Debug Threshold value is more restrictive and discards communication logs of lower order than the one set in the system.

Отладка уровней (от нижних к верхним)

- Отладка
- Информация
- Уведомление
- Ошибка

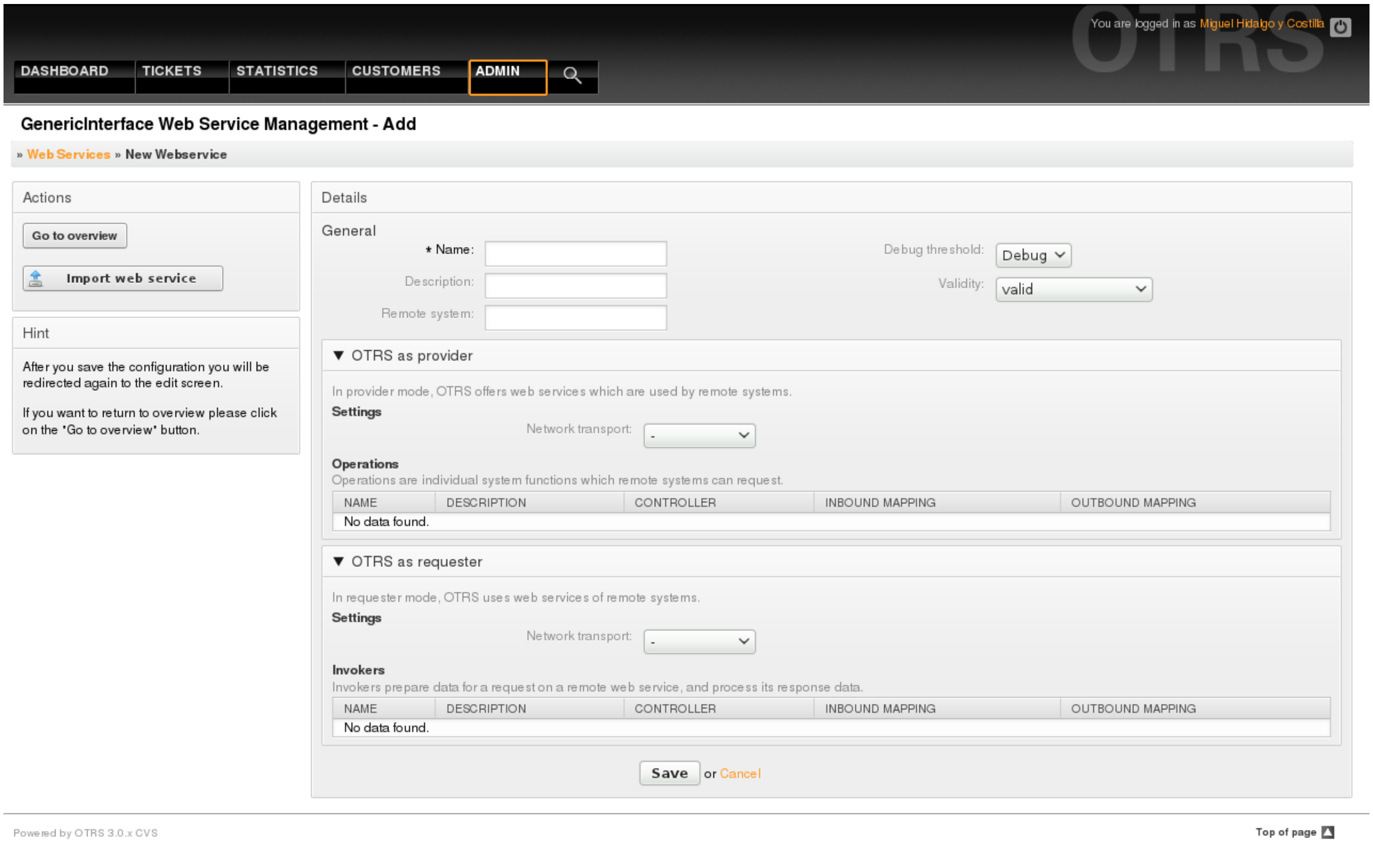
Также есть возможность определить протокол транспортной сети для "OTRS как Поставщика" и "OTRS как Запрашиваемую Сторону".

Нажмите кнопку "Сохнанить", чтобы зарегистрировать новый веб-сервис в базе данных или кнопку "Отменить" чтобы отменить эту операцию.

If you already have a web service configuration file in YAML format you can click on the "Import web service" button on the left side of the screen. For more information on importing web services please check the next section "Web Service Change".

Примечание

Нажмите на имя веб-сервиса на странице просмотра чтобы изменить или добавить больше элементов веб-сервису.



GenericInterface Web Service Management - Add

» Web Services » New Webservice

Actions

Hint

After you save the configuration you will be redirected again to the edit screen.

If you want to return to overview please click on the "Go to overview" button.

Details

General

* Name:	
Description:	
Remote system:	
Debug threshold:	Debug
Validity:	valid

▼ OTRS as provider

In provider mode, OTRS offers web services which are used by remote systems.

Settings

Network transport: -

Operations

Operations are individual system functions which remote systems can request.

NAME	DESCRIPTION	CONTROLLER	INBOUND MAPPING	OUTBOUND MAPPING
No data found.				

▼ OTRS as requester

In requester mode, OTRS uses web services of remote systems.

Settings

Network transport: -

Invokers

Invokers prepare data for a request on a remote web service, and process its response data.

NAME	DESCRIPTION	CONTROLLER	INBOUND MAPPING	OUTBOUND MAPPING
No data found.				

Рисунок: Добавление нового веб-сервиса

11.4.3. Изменения веб-сервиса:

On this screen you have a complete set of functions to handle every part of a web service. On the left side in the action column you can find some buttons that allows you to perform all possible actions on a web service:

- Клонирование веб-сервиса.
- Экспортировать веб-сервис.
- Импортировать веб-сервис.
- История конфигурации.
- Удалить веб-сервис.
- Отладчик.

Примечание

Переход к "История конфигурации" и "Отладчик" приведет к открытию различных страниц.

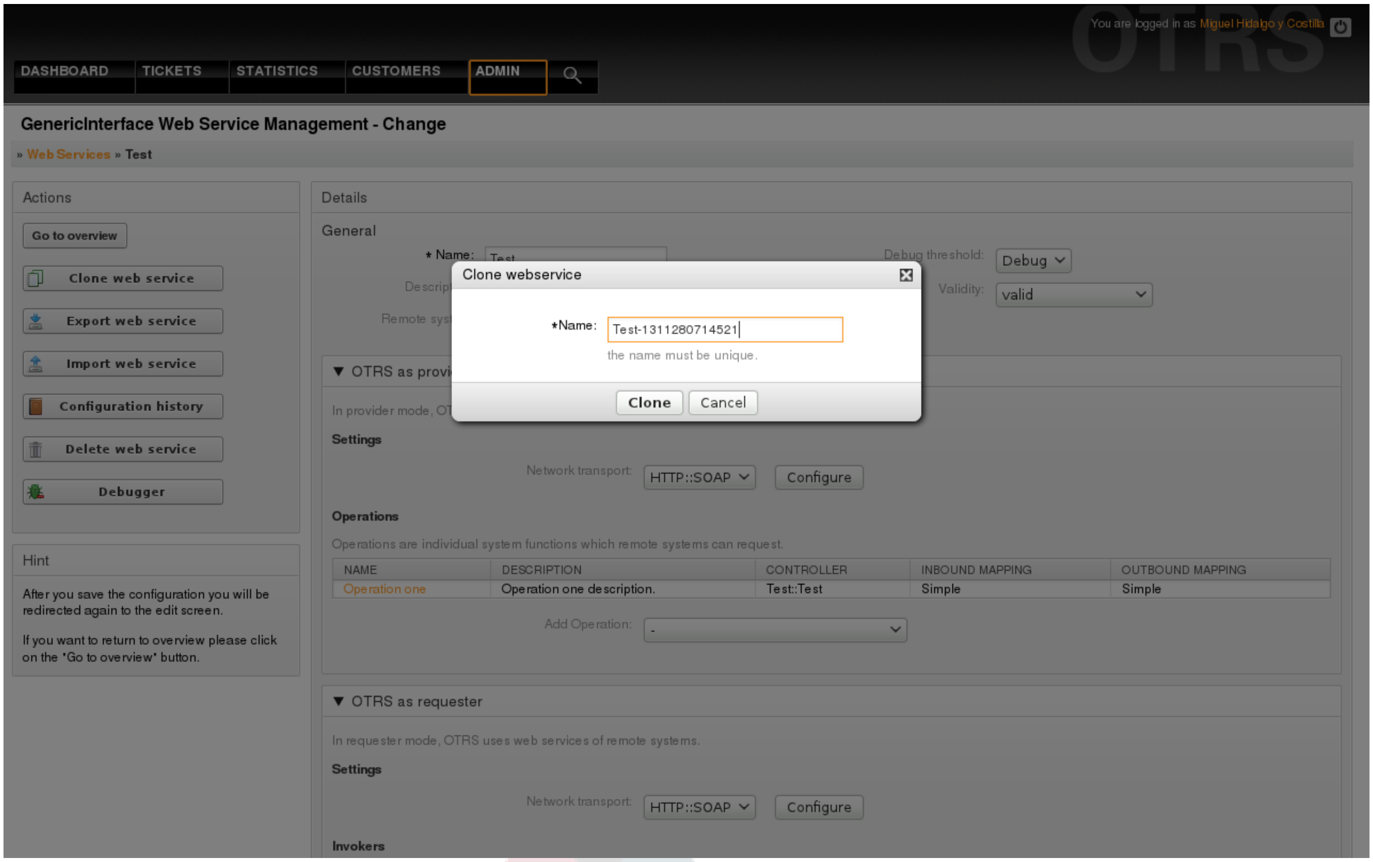
11.4.3.1. Клонирование веб-сервиса:

To clone a web service, you need to click on the "Clone web service" button. A dialog will be shown where you can use the default name or set a new name for the (cloned) web service.

Примечание

Remember that the name of the web service must be unique within the system.

Нажмите кнопку "Клонировать" чтобы создать копию веб-сервиса или кнопку "Отменить", чтобы закрыть диалоговое окно.



GenericInterface Web Service Management - Change

Hint

After you save the configuration you will be redirected again to the edit screen.

If you want to return to overview please click on the "Go to overview" button.

* Name: Test

Debug threshold: Debug

Validity: valid

*Name: Test-1311280714521

the name must be unique.

▼ OTRS as provi

In provider mode, OT

Settings

Network transport: HTTP::SOAP

Operations

Operations are individual system functions which remote systems can request.

NAME	DESCRIPTION	CONTROLLER	INBOUND MAPPING	OUTBOUND MAPPING
Operation one	Operation one description.	Test::Test	Simple	Simple

Add Operation: -

▼ OTRS as requester

In requester mode, OTRS uses web services of remote systems.

Settings

Network transport: HTTP::SOAP

Рисунок: Клонирование веб-сервиса.

11.4.3.2. Экспорт веб-сервиса

The "Export web service" button gives you the opportunity to dump the configuration of the current web service into a YAML file, to download it and to store it on your file system. This can be specially useful if you want to migrate the web service from one server to another, for example from a testing environment to a production system.

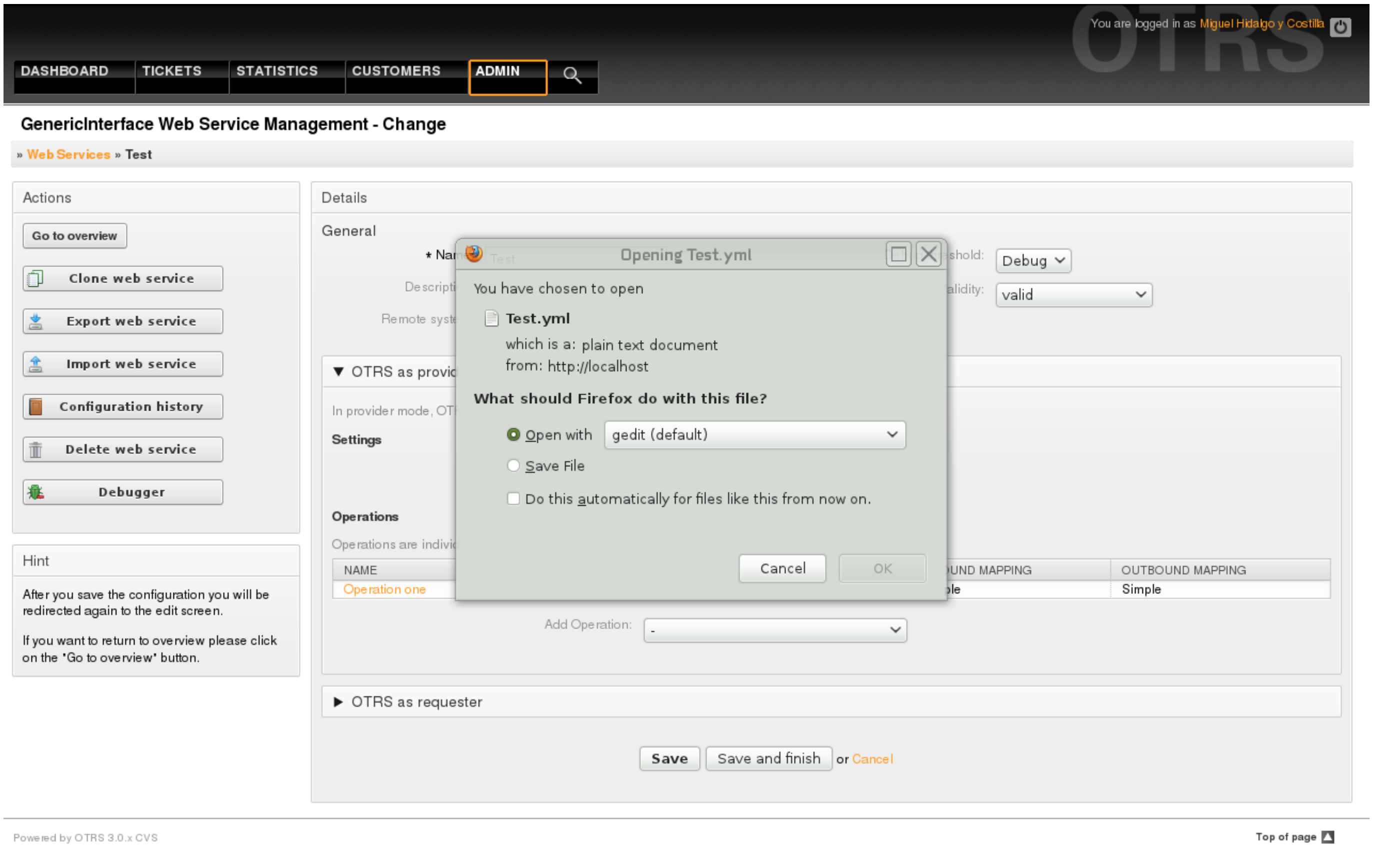
Предупреждение

All stored passwords in the web service configuration will be exported in plain text format.

Сразу же после нажатия кнопки "Экспорт веб-сервиса" появится диалоговое окно сохранения, на подобе того, которое появляется когда вы нажимаете в браузере ссылку сохранить.

Примечание

Each browser on each operating system has its own save dialog screen and style. Depending on the browser and its configuration it is possible that no dialog is shown and the file is saved to a default directory on your file system. Please check your browser documentation for more specific instructions if needed.



GenericInterface Web Service Management - Change

» Web Services » Test

Hint

After you save the configuration you will be redirected again to the edit screen.

If you want to return to overview please click on the "Go to overview" button.

Opening Test.yml

You have chosen to open

Test.yml

which is a: plain text document
from: http://localhost

What should Firefox do with this file?

● Open with gedit (default)

○ Save File

□ Do this automatically for files like this from now on.

Рисунок: Экспорт веб-сервиса.

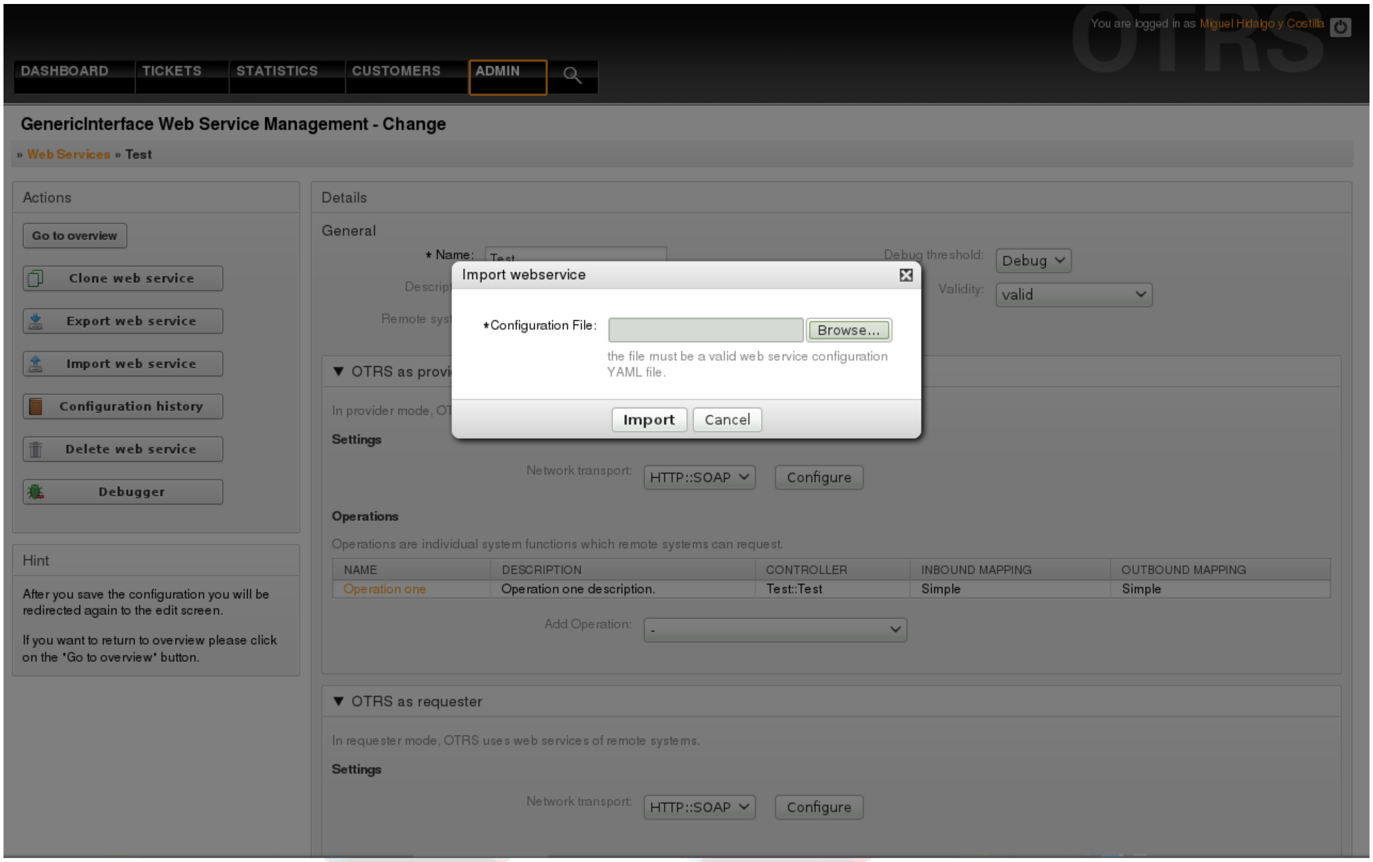
11.4.3.3. Импорт веб-сервиса:

A valid web service configuration YAML file is required to use the import web service feature. Click on the "Import web service" button, browse for the configuration file or provide the complete path in the input box.

Нажмите кнопку "Импорт" чтобы создать новый веб-сервис с файла или кнопку "Отменить" чтобы закрыть диалоговое окно.

Примечание

The web service name will be taken from the configuration file name (e.g. if the file name is MyWebservice.yml the resulting web service will be named MyWebservice). If a web service is registered in the system with the same name as the web service that you want to import, the system will lead you to the web service change screen to let you change the name of the imported web service.



GenericInterface Web Service Management - Change

» Web Services » Test

Actions

Hint

After you save the configuration you will be redirected again to the edit screen.

If you want to return to overview please click on the "Go to overview" button.

Details

General

* Name: Test

Debug threshold: Debug

Validity: valid

Import webservice

*Configuration File:

the file must be a valid web service configuration YAML file.

OTRS as provi

In provider mode, OT

Settings

Network transport: HTTP::SOAP

Operations

Operations are individual system functions which remote systems can request.

NAME	DESCRIPTION	CONTROLLER	INBOUND MAPPING	OUTBOUND MAPPING
Operation one	Operation one description.	Test::Test	Simple	Simple

Add Operation: -

OTRS as requester

In requester mode, OTRS uses web services of remote systems.

Settings

Network transport: HTTP::SOAP

Рисунок: Импорт веб-сервиса.

11.4.3.4. История веб-сервиса

Every change to the web service configuration creates a new entry in the web service history (as a journal). The web service history screen displays a list of all configuration versions for a web service. Each row (version) in the "Configuration History List" represents a single revision in the web service history.

Click on one of the rows to show the whole configuration as it was on that particular date / time. The configuration will be shown in the "History details" section of this screen. Here you are also able to export the selected web service configuration version or to restore that version into the current web service configuration.

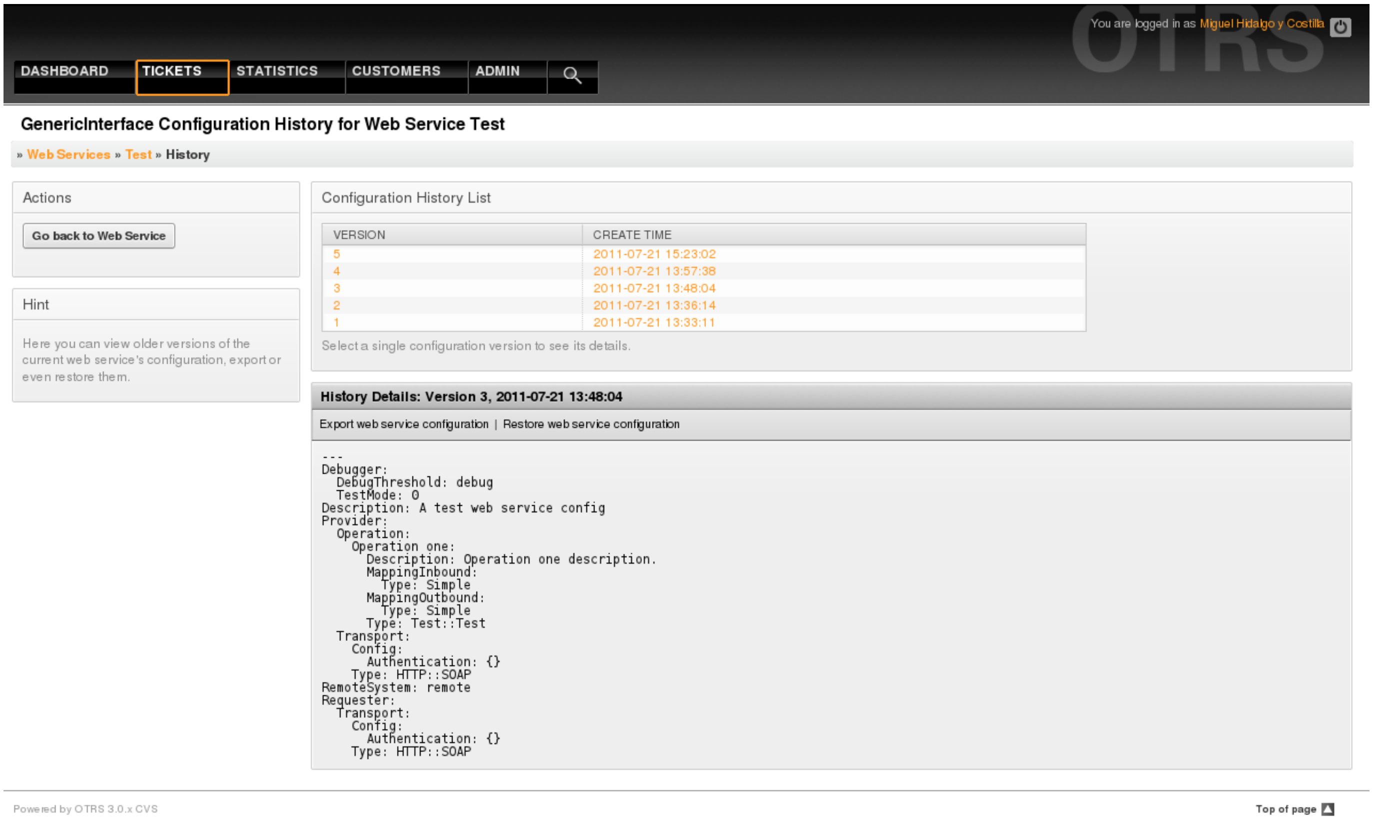
"Экспортировать настройки веб-сервиса" ведет себя точно так же как "Экспортировать веб-сервис". Для более подробной информации обратитесь к этому разделу.

If changes to the current web service configuration do not work as expected and it is not easy to revert the changes manually, you can click on the "Revert web service configuration" button. This will open a dialog to ask you if you are sure to revert the web service configuration. Click "Revert web service configuration" in this dialog to replace the current configuration with the selected version, or click "Cancel" to close the dialog.

Предупреждение

Remember that any passwords stored in the web service configuration will be exported in plain text format.

Please be careful when you restore a configuration because this the process is irreversible.



GenericInterface Configuration History for Web Service Test

» Web Services » Test » History

Actions

Hint

Here you can view older versions of the current web service's configuration, export or even restore them.

Configuration History List

VERSION	CREATE TIME
5	2011-07-21 15:23:02
4	2011-07-21 13:57:38
3	2011-07-21 13:48:04
2	2011-07-21 13:36:14
1	2011-07-21 13:33:11

Select a single configuration version to see its details.

History Details: Version 3, 2011-07-21 13:48:04

Export web service configuration | Restore web service configuration

```
---
Debugger:
  DebugThreshold: debug
  TestMode: 0
Description: A test web service config
Provider:
  Operation:
    Operation one:
      Description: Operation one description.
      MappingInbound:
        Type: Simple
      MappingOutbound:
        Type: Simple
      Type: Test::Test
  Transport:
    Config:
      Authentication: {}
    Type: HTTP::SOAP
RemoteSystem: remote
Requester:
  Transport:
    Config:
      Authentication: {}
    Type: HTTP::SOAP
```

Рисунок: История веб-сервиса

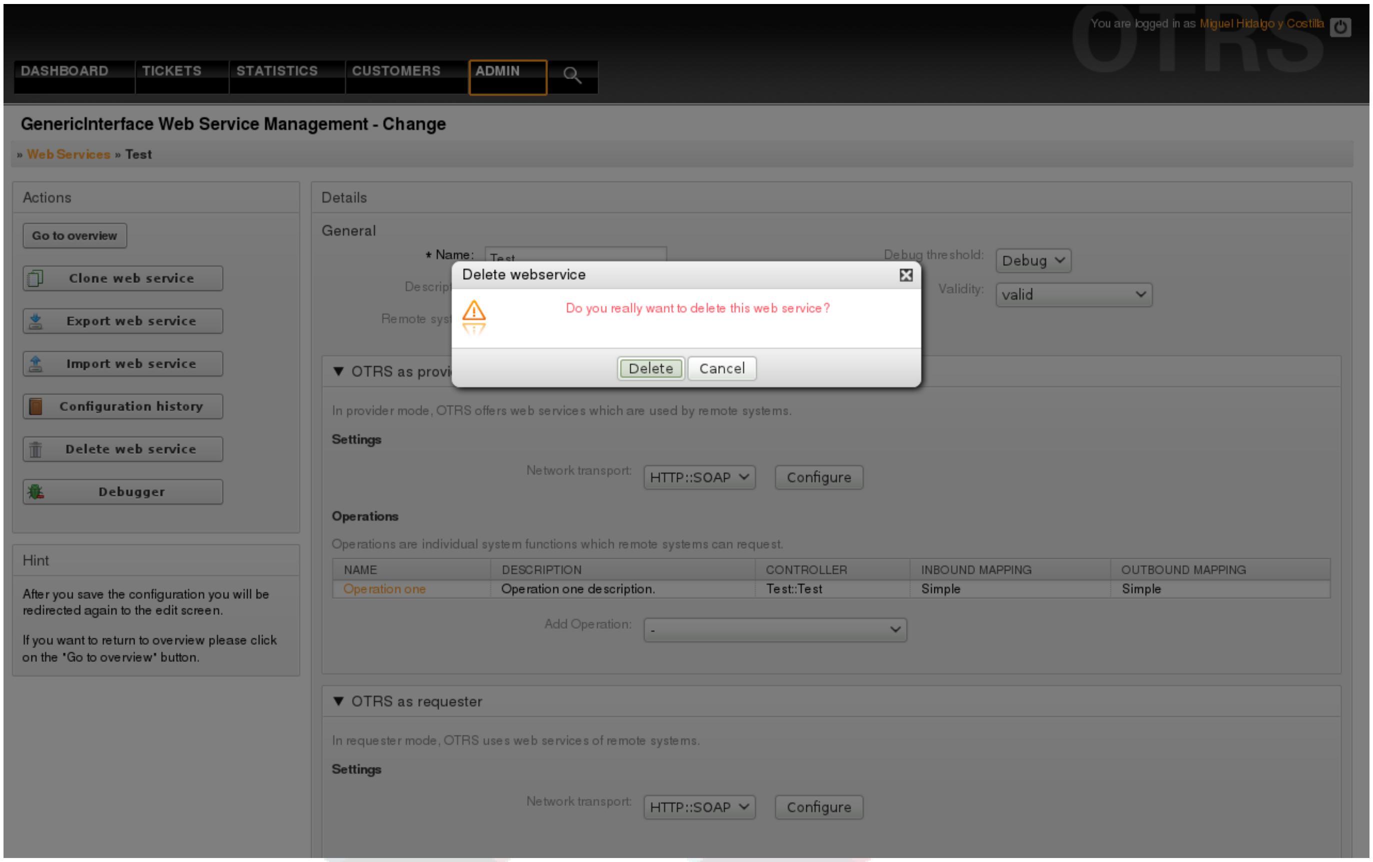
11.4.3.5. Удаление веб-сервиса

Иногда возникает необходимость удалить веб-сервис полностью. Для этого можно нажать кнопку "Удалить веб-сервис", появится диалоговое окно с вопросом о подтверждении операции.

Click on "Delete" to confirm the removal of the web service or on "Cancel" to close the dialog.

Предупреждение

Deleting a web service can't be undone, please be careful when deleting a web service.



You are logged in as Miguel Hidalgo y Costilla

GenericInterface Web Service Management - Change

» Web Services » Test

Actions

Hint

After you save the configuration you will be redirected again to the edit screen.

If you want to return to overview please click on the "Go to overview" button.

Details

General

* Name: Test

Debug threshold: Debug

Validity: valid

Delete webservice

Do you really want to delete this web service?

OTRS as provider

In provider mode, OTRS offers web services which are used by remote systems.

Settings

Network transport: HTTP::SOAP

Operations

Operations are individual system functions which remote systems can request.

NAME	DESCRIPTION	CONTROLLER	INBOUND MAPPING	OUTBOUND MAPPING
Operation one	Operation one description.	Test::Test	Simple	Simple

Add Operation: -

OTRS as requester

In requester mode, OTRS uses web services of remote systems.

Settings

Network transport: HTTP::SOAP

Рисунок: Удаление веб-сервиса

11.4.3.6. Отладчик веб-сервиса:

Отладчик хранит журнал веб-сервиса. На странице отладчика можно следить за всеми связями веб-сервера, когда он работает в роли поставщика и получателя.

Когда отображается эта страница начинает загружаться список запросов. После того как список будет полность загружен, можно выбрать одну из строк (что означает последовательность связей) и просмотреть подробню информацию. Эта информация будет отображаться во всплывающем окне ниже.

Можно сузить список используя фильтр в правой части страницы. Фильтрацию можно производить по:

- Тип связи (поставщик или запрашиваемая сторона)
- Дата: до и / или после определенной даты
- Удаленный IP-адрес
- Сочетание всех.

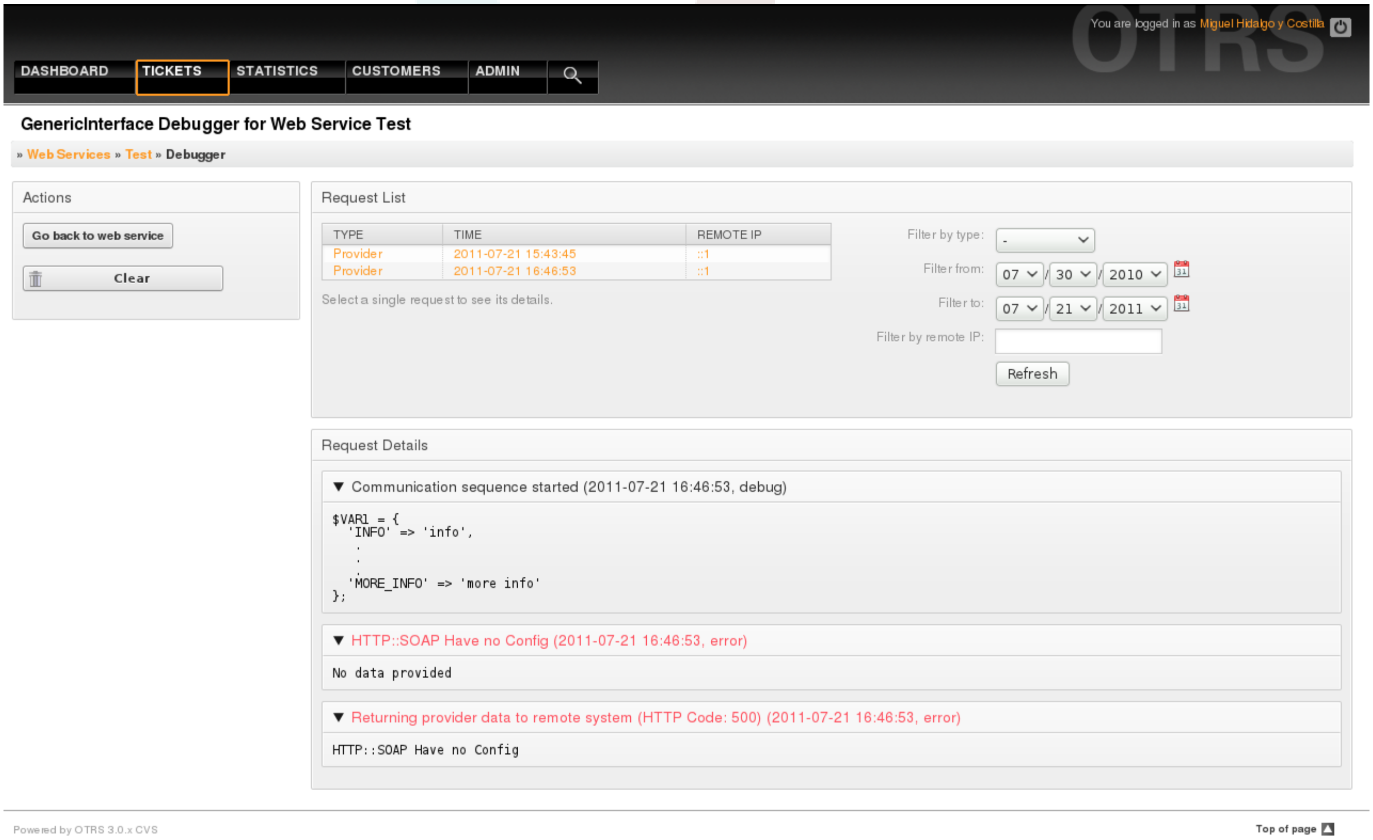
После того как настройки фильтра завершены, нажмите кнопку "Обновить" и новый список, отвечающий вашим критериям будет отображен.

Примечание

В зависимости от критериев поиска установленых в фильтре, новый поиск может вернуть пустой список.

On the left part of the screen under the action column you can select "Go back to the web service" or clear the debugger log by pushing the "Clear" button. This will open a dialog that ask you to confirm erasing of the log. Click "Clear" in the dialog button to perform the action or click on "Cancel" to close this dialog.

В разделе "Подробности Запроса" можно просматривать подробную информацию о комуникации. Здесь можно следить за полным потоком и проверять всевозможные ошибки или подтвержать успешные ответы.



GenericInterface Debugger for Web Service Test

» Web Services » Test » Debugger

Actions

Request List

TYPE	TIME	REMOTE IP
Provider	2011-07-21 15:43:45	::1
Provider	2011-07-21 16:46:53	::1

Select a single request to see its details.

Filter by type: -

Filter from: 07 / 30 / 2010

Filter to: 07 / 21 / 2011

Filter by remote IP:

Request Details

▼ Communication sequence started (2011-07-21 16:46:53, debug)

```
$VAR1 = {
  'INFO' => 'info',
  .
  .
  'MORE_INFO' => 'more info'
};
```

▼ HTTP::SOAP Have no Config (2011-07-21 16:46:53, error)

```
No data provided
```

▼ Returning provider data to remote system (HTTP Code: 500) (2011-07-21 16:46:53, error)

```
HTTP::SOAP Have no Config
```

Рисунок: Отладчик веб-сервиса.

11.4.3.7. Изменение настроек веб-службы

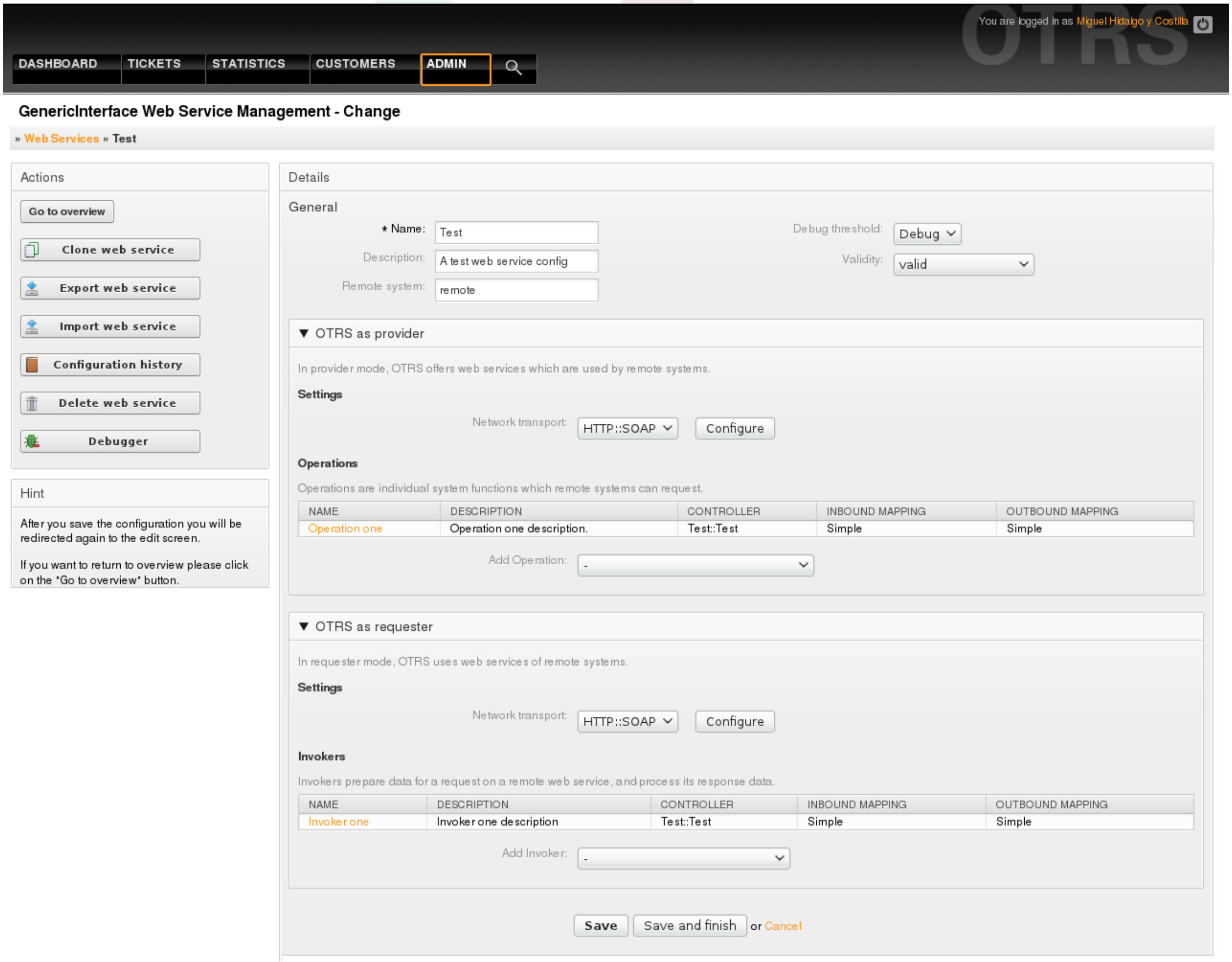
Returning to the web service change screen, now we are going to review the right side of it. Here we have the possibility to modify all the general data for a web service such as name, description, debug threshold, etc. Also there are two more sections below that allows us to modify specific parameters for communication types "OTRS as Provider" and "OTRS as Requester".

The web service configuration needs to be saved on each level. This means that if a setting is changed, links to other, deeper parts of the configuration will be disabled forcing you to save the current configuration level. After saving the disabled links will be re-enabled again allowing you to continue with the configuration.

On the "OTRS as provider" section it is possible to set or configure the network transport protocol. Only network transport back-ends that are registered are shown on the list. To configure the network transport click on the "Configure" button. It is also possible to add new operations in this box. To do this select one of the available operations from the "Add Operation" list. This will lead you to the operation configuration screen. After saving the new operation it will be listed in the table above.

"OTRS как запрашивающая сторона" очень похож на предыдущий, но вместо "операций" здесь можно добавить invokers

Click the "Save" button to save and continue configuring the web service, "Save and finish" to save and return to the web service overview screen, or "Cancel" to discard current configuration level changes and return to web service overview screen.



GenericInterface Web Service Management - Change

» Web Services » Test

After you save the configuration you will be redirected again to the edit screen.

If you want to return to overview please click on the "Go to overview" button.

* Name:	Test
Description:	A test web service config
Remote system:	remote
Debug threshold:	Debug
Validity:	valid

▼ OTRS as provider

In provider mode, OTRS offers web services which are used by remote systems.

Settings

Network transport: HTTP::SOAP Configure

Operations

Operations are individual system functions which remote systems can request.

NAME	DESCRIPTION	CONTROLLER	INBOUND MAPPING	OUTBOUND MAPPING
Operation one	Operation one description.	Test::Test	Simple	Simple

Add Operation: -

▼ OTRS as requester

In requester mode, OTRS uses web services of remote systems.

Settings

Network transport: HTTP::SOAP Configure

Invokers

Invokers prepare data for a request on a remote web service, and process its response data.

NAME	DESCRIPTION	CONTROLLER	INBOUND MAPPING	OUTBOUND MAPPING
Invoker one	Invoker one description	Test::Test	Simple	Simple

Add Invoker: -

Рисунок: Изменения веб-службы.

Примечание

Like the other Generic Interface configuration screens such as Network Transport, Operation, Invoker and Mapping, the initial configuration (add) screen will only present two options: "Save" and "Cancel". If the configuration is re-visited, a new option "Save and Finish" will appear. The behavior of this feature is defined below.

"Save" will store the current configuration level in the database and it will return to the previous screen to review your changes or to configure deeper settings.

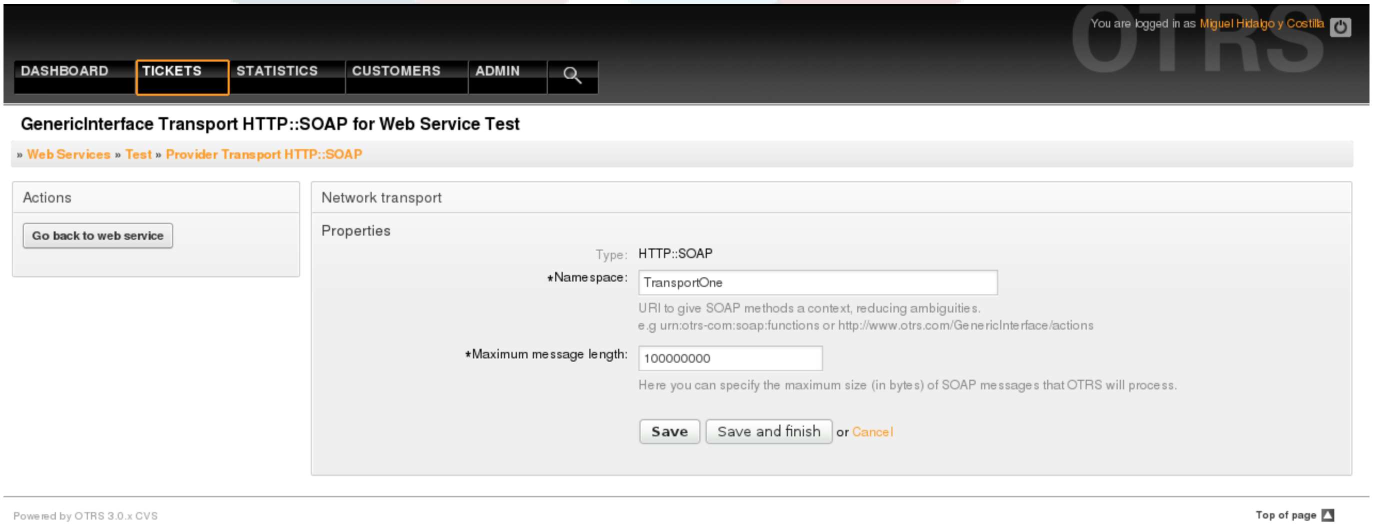
"Save and Finish" сохнанит текущий уровень настроек в базе данных и вернется к предыдущей странице в иерархии конфигурации.

"Отменить" будет отменять любые изменения конфигурации и возвращать их к текущему уровню конфигурации, после чего возвращаться на к предыдущему экрану в иерархии настроек.

11.4.3.7.1. Транспортаня сеть Веб-сервиса

In future the list of available network transports will be increased. Currently only the "HTTP::SOAP" transport is available. Each transport has different configuration options to setup and they might use different frontend modules to configure it, but mostly they should look similar to the "HTTP::SOAP" transport configuration module.

It is quite simple to configure the "HTTP::SOAP" protocol as provider. There are only two settings: "Namespace" and "Maximum message length". These fields are required. The first one is a URI to give SOAP methods a context, reducing ambiguities, and the second one is a field where you can specify the maximum size (in bytes) for SOAP messages that OTRS will process.



GenericInterface Transport HTTP::SOAP for Web Service Test

» Web Services » Test » Provider Transport HTTP::SOAP

Actions

Go back to web service

Network transport

Properties

Type: HTTP::SOAP

*Namespace: TransportOne

URI to give SOAP methods a context, reducing ambiguities.
e.g urn:otrs-com:soap:functions or http://www.otrs.com/GenericInterface/actions

*Maximum message length: 100000000

Here you can specify the maximum size (in bytes) of SOAP messages that OTRS will process.

Save | Save and finish | or Cancel

Powered by OTRS 3.0.x CVS

Top of page

Рисунок: Транспортаня сеть Веб-сервиса.

11.4.3.7.2. Операции веб-сервиса

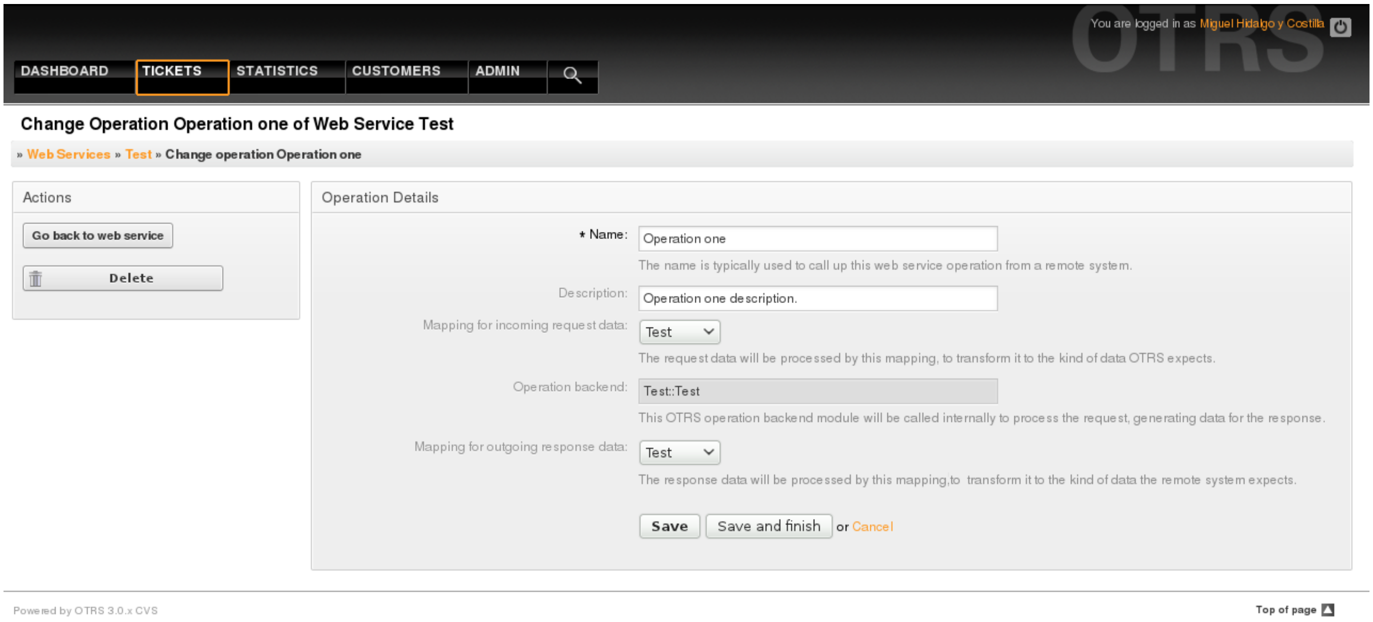
The actions that can be performed when you are using OTRS as a provider are called "Operations". Each operation belongs to a controller. Controllers are collections of operations or invokers, normally operations from the same controller need similar settings and share the same configuration dialog. But each operation can have independent configuration dialogues if needed.

Name, Description, Backend, and Mappings are fields that normally appear on every operation, other special fields can appear in non default configuration dialogues to fulfill specific needs of the operation.

Normally there are two mapping configuration sections on each operation, one for the incoming data and another one for the outgoing data. You can choose different mapping types (backends) for each mapping direction, since their configuration is independent from each other and also independent from the operation backend. The normal and most common practice is that the operation uses the same mapping type in both cases (with inverted configuration). The complete mapping configuration is done in a separate screen which depends on the mapping type.

The operation backend is pre-populated and is not editable. You will see this parameter when you choose the operation on the web service edit screen. The field is only informative.

In the left part of the screen on the action column you have the options: "Go back to web service" (discarding all changes since the last save) and "Delete". If you click on the last one, a dialog will open and ask you if you like to remove the operation. Click on "Delete" button to confirm the removal of the operation and it configuration or "Cancel" to close the delete dialog.



Change Operation Operation one of Web Service Test

» Web Services » Test » Change operation Operation one

Actions

Operation Details

* Name: Operation one

The name is typically used to call up this web service operation from a remote system.

Description: Operation one description.

Mapping for incoming request data: Test

The request data will be processed by this mapping, to transform it to the kind of data OTRS expects.

Operation backend: Test::Test

This OTRS operation backend module will be called internally to process the request, generating data for the response.

Mapping for outgoing response data: Test

The response data will be processed by this mapping,to transform it to the kind of data the remote system expects.

Рисунок: Операции веб-сервиса

11.4.3.7.3. Web Service Requester Network Transport

The network transport configuration for the requester is similar to the configuration for the provider. For the Requester "HTTP::SOAP" network transport there are more fields to be set.

Apart from the "Endpoint" (URI of the Remote System web service interface to accept requests) and "Namespace" which are required fields, you can also specify:

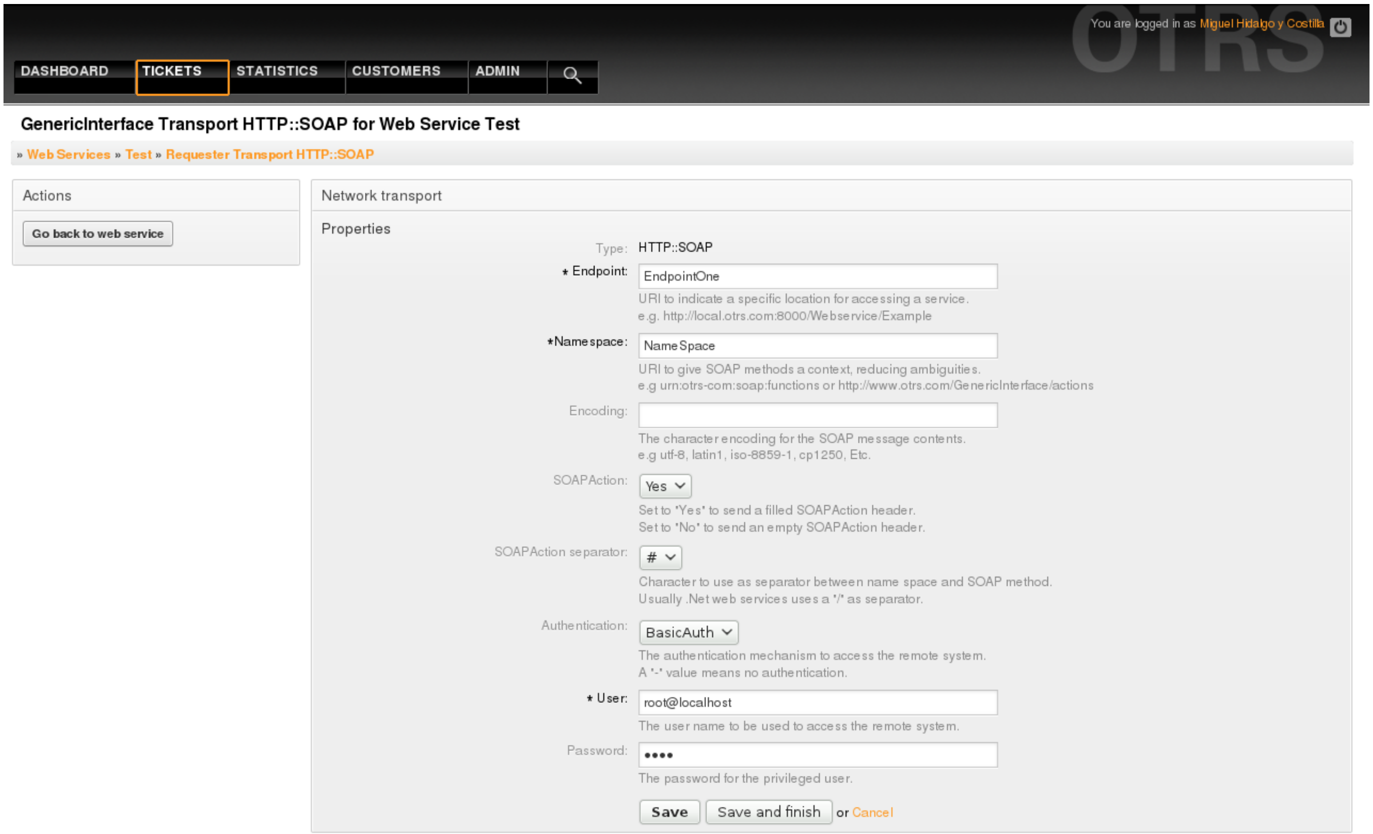
- Кодировка (такая как utf-8, latin1, iso-8859-1, cp1250, и т.д.) для SOAP-сообщений.
- SOAPAction Header: you can use this to send an empty or filled SOAPAction header. Set to "No" and the SOAPAction header on the SOAP message will be an empty string, or set to "Yes" to send the soap action in Namespace#Action format and define the separator (typically "/" for .Net web services and "#" for the rest).
- Аутентификация: установить механизм аутентификации, установлено в "-" чтобы не использовать любую проверку подлинности или выбрать ее из списка, после чего появится поле с подробной информацией.

Примечание

Currently only the "BasicAuth" (HTTP) authentication mechanism is implemented. You can decide whether or not to use it depending on the Remote System configuration. If used, you must provide the User Name and the Password to access the remote system.

Предупреждение

If you supply a password for authentication and after you export the web service to a YAML file this password will be revealed and will be written into a plain text string inside the YAML file. Be aware of it and take precautions if needed.



GenericInterface Transport HTTP::SOAP for Web Service Test

» Web Services » Test » Requester Transport HTTP::SOAP

Network transport

Properties

Type: HTTP::SOAP

* Endpoint: EndpointOne
URI to indicate a specific location for accessing a service.
e.g. http://local.otrs.com:8000/Webservice/Example

*Namespace: NameSpace
URI to give SOAP methods a context, reducing ambiguities.
e.g urn:otrs-com:soap:functions or http://www.otrs.com/GenericInterface/actions

Encoding:
The character encoding for the SOAP message contents.
e.g utf-8, latin1, iso-8859-1, cp1250, Etc.

SOAPAction: Yes
Set to "Yes" to send a filled SOAPAction header.
Set to "No" to send an empty SOAPAction header.

SOAPAction separator: #
Character to use as separator between name space and SOAP method.
Usually .Net web services uses a "/" as separator.

Authentication: BasicAuth
The authentication mechanism to access the remote system.
A "-" value means no authentication.

* User: root@localhost
The user name to be used to access the remote system.

Password: ••••
The password for the privileged user.

Figure: Web service requester network transport.

11.4.3.7.4. Вызов веб-сервиса:

The actions that can be performed when you are using OTRS as a requester are called "Invokers". Each invoker belongs to a controller (controllers are collections of operations or invokers). Usually invokers from the same controller need similar settings and share the same configuration dialogues. Each invoker can have independent configuration dialogues if needed.

Name, Description, Backend, and Mappings are fields that normally appear on every invoker. Additionally the list of event triggers and other special fields can appear on non default configuration dialogues to fulfill special needs of the invoker.

Normally there are two mapping configuration sections for each invoker, one for the incoming data and another one for the outgoing data. You can choose different mapping types (backends) for each mapping direction, since their configuration is independent from each other and also independent from the invoker backend. The normal and most common practice is that the invoker uses the same mapping type in both cases, with inverted configuration. The complete mapping configuration is done in a separate screen, which depends on the mapping type.

The invoker backend is pre-populated and can not be edited. You will see this parameter when you choose the invoker on the web service edit screen. The field is only informative.

Event triggers are events within OTRS such as "TicketCreate", "ArticleSend", etc. These can act as triggers to execute the invoker. Each invoker needs to have at least one event trigger registered, or the invoker will be useless, because it will never be called. The asynchronous property of the event triggers define if the OTRS process will handle the invoker or if it will be delegated to the Scheduler.

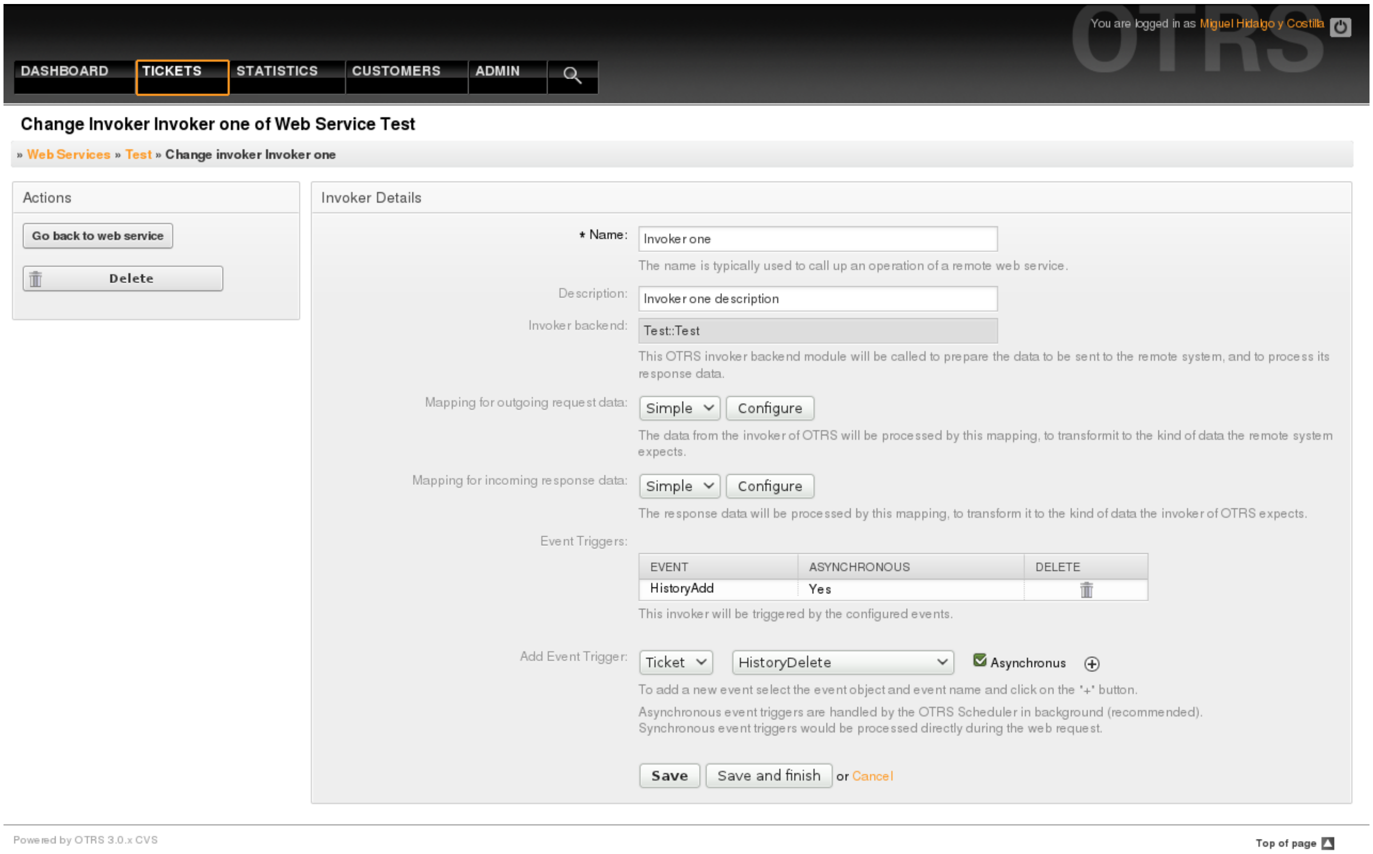
Примечание

The OTRS Scheduler is a separate process that executes tasks in the background. Using this the OTRS process itself will not be affected if the Remote System takes a long time to respond, if it is not available or if there are network problems. If you don't use the scheduler using web services can make OTRS slow or non-responsive. Therefore it is highly recommend to use asynchronous event triggers as often as possible.

To add an Event trigger, first select the event family from the first list, then the event name from the second list, then set the asynchronous property (if unchecked means that the event trigger will not be asynchronous) and finally click on the plus button. A new event trigger will be created and it will be listed on the invoker "Event Triggers" list.

To delete an Event trigger, simply locate the event trigger to be deleted in the "Event Triggers" list and click on the trash icon at the end of the row. This will open a dialog that asks you if you are sure to delete the event trigger. Click "Delete" to remove the event trigger from the list, or "Cancel" to close the dialog.

In the left part of the screen on the action column you have the options: "Go back to web service" (discarding all changes since the last save) and "Delete". If you click on the last one, a dialog will emerge and ask you if you like to remove the invoker. Click on the "Delete" button to confirm the removal of the invoker and its configuration or "Cancel" to close the delete dialog.



Change Invoker Invoker one of Web Service Test

» Web Services » Test » Change invoker Invoker one

Invoker Details

* Name: Invoker one

The name is typically used to call up an operation of a remote web service.

Description: Invoker one description

Invoker backend: Test::Test

This OTRS invoker backend module will be called to prepare the data to be sent to the remote system, and to process its response data.

Mapping for outgoing request data: Simple Configure

The data from the invoker of OTRS will be processed by this mapping, to transformit to the kind of data the remote system expects.

Mapping for incoming response data: Simple Configure

The response data will be processed by this mapping, to transform it to the kind of data the invoker of OTRS expects.

Event Triggers:

EVENT	ASYNCHRONOUS	DELETE
HistoryAdd	Yes	🗑

This invoker will be triggered by the configured events.

Add Event Trigger: Ticket HistoryDelete ☑ Asynchronus ⊕

To add a new event select the event object and event name and click on the '+' button.

Asynchronous event triggers are handled by the OTRS Scheduler in background (recommended).
Synchronous event triggers would be processed directly during the web request.

Save Save and finish or Cancel

Figure: Web service invoker.

11.4.3.7.5. Web Service Mapping

There are cases where you need to transform the data from one format to another (map or change data structure), because normally a web service is used to interact with a Remote System, that is highly probable that is not another OTRS system and / or could not understand the OTRS data structures and values. In these cases some or all values have to be changed, and sometimes even the names of the values (keys) or even the complete structure, in order to match with the expected data on the other end. To accomplish this task the Generic Interface Mapping Layer exists.

Each Remote System has it own data structures and it is possible to create new mapping modules for each case (e.g. there is a customized mapping module for SAP Solution Manager shipped with OTRS), but it is not always necessary. The module Mapping::Simple should cover most of the mapping needs.

Примечание

When Mapping::Simple does not cover all mapping needs for a web service, a new mapping module should be created. To learn more about how to create new mapping modules please consult the OTRS Development Manual.

Этот модуль дает возможность установить значение для отображения по умолчанию для каждого ключа или значения для всех передаваемых данных.

At the beginning of the screen you will see a general section where you can set the default rules that will apply for all the unmapped keys and values. There are three options available, these options are listed below:

- Не изменять: не касается ключей и значений в любом случае.

- Ignore (drop key/value pair): when this is applied to the key it deletes the key and value, because when a key is deleted then in consequence its associated value is deleted too. When this is applied to the value, only the value is deleted, keeping the key, that now will be associated to an empty value.

- MapTo (use provided key or value as default): all keys and / or values without a defined map rule, will use this as default, when you select this option a new text field will appear to set this default.

Clicking on the "+" button for new key map, will display a new box for a single mapping configuration. You can add as many key mappings as needed. Just click on the "+" button again and a new mapping box will appear below the existing one. From this mapping boxes you can define a map for a single key, with the next options:

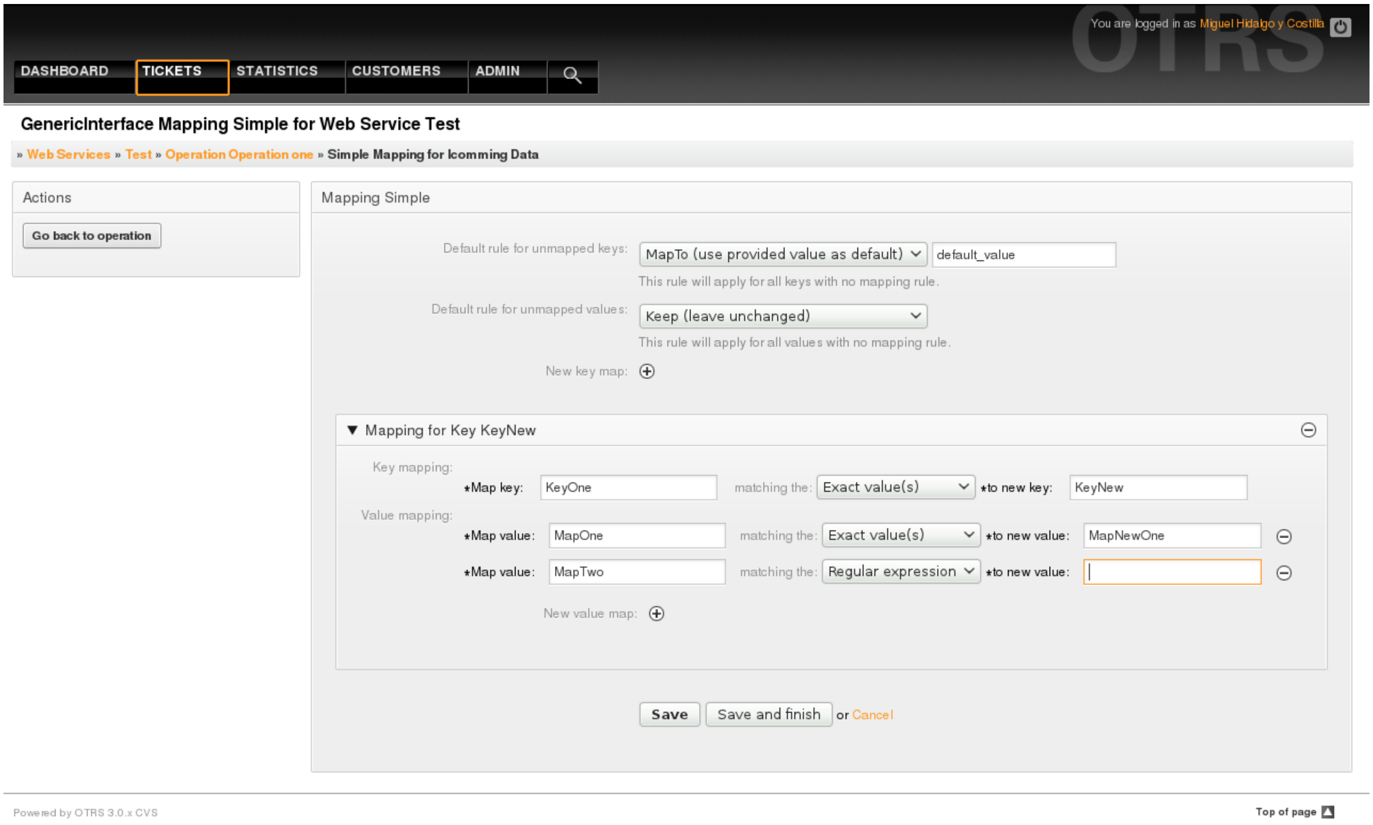
- Точное значение(я): строка со старым значением будет заменена новой в случае полного совпадения "старого" ключа.

- Регулярное выражение: строки будут изменены с помощью правила преобразования.

Pressing the new value map "+" button will display a new row for a value map. Here it is also possible to define rules for each value to be mapped with the same options as for the key map (Exact value and Regular expression). You can add as many values to map as needed, and if you want to delete one of them, just click on the "-" button for each mapping value row.

Удаление целого раздела ключей отображения возможно, просто нажмите на кнопку "-", расположеную в верхнем правом углу каждого окна, для которого вы хотите удалить.

If you need to delete a complete mapping configuration: go back to the corresponding operation or invoker screen, look for the mapping direction that you select before and set its value to "-", and save the configuration to apply changes.



GenericInterface Mapping Simple for Web Service Test

» Web Services » Test » Operation Operation one » Simple Mapping for Icomming Data

Mapping Simple

Default rule for unmapped keys: MapTo (use provided value as default) | default_value
This rule will apply for all keys with no mapping rule.

Default rule for unmapped values: Keep (leave unchanged)
This rule will apply for all values with no mapping rule.

New key map:

Mapping for Key KeyNew

Key mapping:
*Map key: KeyOne matching the: Exact value(s) *to new key: KeyNew

Value mapping:
*Map value: MapOne matching the: Exact value(s) *to new value: MapNewOne
*Map value: MapTwo matching the: Regular expression *to new value:

New value map:

Figure: Web service mapping.

11.5. Интерфейс командной строки Веб-Сервиса

The Command Line Interface (CLI) is a fast way to work with the web services. It consists of a set of tools that can be use to perform basic operations like:

- Создать, Обновить, Читать, Просматривать и Удалить веб-сервисы основаные основаные на YAML-файлах.
- Чтение журнала Отладчика с опциями фильтрации.

Примечание

You don't need to use the CLI to work with web services. Integrated into the Admin interface there is a complete set of screens to interact with every part of the web services. Please read the web service GUI section included in this manual.

11.5.1. Конфигурация Веб-сервиса

Файл "WebserviceConfig.pl" был разработан для того чтобы создать простой, но в то же время быстрый и мощный инструмент для работы с настройками веб-сервиса. Он предоставляет возможность выполнять следующие действия:

- Add: для создания веб-сервисов с использованием YAML-файлов в качестве источника конфигурации.
- Update: изменить существующий веб-сервис, конфигурация может быть изменена с использованием других или измененных YAML-файлов.

- Read: для вывода на экран текущих настроек веб-сервиса.
- List: чтобы получить полный список всех веб-сервисов, зарегистрированных в системе.
- Delete: для удаления веб-сервиса из системы. Будьте осторожны при выполнении этой операции, ее невозможно отменить.

Предупреждение

Операция веб-сервиса READ будет отображать на экране все настройки в виде простого текста, включаяя сохраненные пароли. Помните об этом и соблюдайте необходимые меры предосторожности!

Пример: Создание новой конфигурации веб-сервиса:

```
                shell> OTRS_HOME/bin/otrs.WebserviceConfig.pl -a write -n <webservice_name> -
f /path/to/yaml/file
```

Также можно использовать файл 'otrs.WebserviceConfig.pl' со следующими опциями:

- **-a read -i <webservice_id>** - Для чтения сохраненной конфигурации.
- **-a write -n <webservice_name> -f /path/to/yaml/file** - Для создания нового веб-сервиса.
- **-a write -i <webservice_id> -f /path/to/yaml/file** - Для обновления веб-сервиса.
- **-a list** - Для просмотра списка доступных веб-сервисов.
- **-a delete -i <webservice_id>** - Для удаления веб-сервиса.

11.5.2. Отладчик веб-сервиса:

Another available tool via the command line interface is the "otrs.GenericInterfaceDebugRead.pl" script, which is an interface to search for web service debugger log entries.

Пример: Поиск записей в журнале отладчика:

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

Дополнительные параметры могут быть использованы для сценария "otrs.GenericInterfaceDebugRead.pl":

- **-c** - для фильтрации по Communication ID (md5sum формат).
- **-t** - для фильтрации по CommunicationType ('Provider' или 'Requester').
- **-a** - для фильтрации по дате (На текущую дату или После нее).
- **-b** - для фильтрации по дате (на текущую дату или До нее).
- **-i** - для фильтрации по IP-адресу (должен быть правильный IPv4 или IPv6 адрес).
- **-w** - для фильтрации по ID веб-сервиса.
- **-d** - включить подробные передаваемые данные.

Пример: Поиск записей журнала отладчика со всеми параметрами:

```
                shell> ./otrs.GenericInterfaceDebugRead.pl -c a7cc4d9f5c70387a9bfbe1351bc88966
-t Provider -a '2011-07-22 00:00:00' -b '2011-07-26 00:00:00' -i 127.0.0.1 -w 123 -d 1
```

Примечание

It is highly recommended to include at least one of the filter options listed above, and even more if the "-d" option is selected, because *a lot of* information can be retrieved from the data base and displayed on the screen, this could result in slow response times and much more information than what you really needed.

11.6. Конфигурация Веб-сервиса

From its design the web services were conceived to be portable from one OTRS system to another, e.g. from a test or development environment to a production system. Therefore it was needed to have an easy way to extract the web service configuration from the database, and import it to another. To accomplish this task the Generic Interface uses YAML files as the web services configuration basis.

Why YAML? YAML is a markup language designed to be human friendly to read and write (it is easier to understand than JSON), it does not have some of the limitations of XML like numeric tags, it is open, standardized, and is complete enough to store the whole web service configuration.

Примечание

Чтобы узнать больше о YAML пожалуйста посетите ссылку http://www.yaml.org/.

Ниже приводится конфигурация веб-сервера на основе файла конфигурации в форматеYAML:

```
---
Debugger:
  DebugThreshold: debug
Description: This an example of a web service configuration
Provider:
  Operation:
    CloseIncident:
      Description: This is a test operation
      MappingInbound: {}
      MappingOutbound: {}
      RemoteSystemGuid: ''
      Type: Test::Test
    Test:
      Description: This is a test operation
      MappingInbound:
        Config:
          KeyMapDefault:
            MapTo: ''
            MapType: Keep
          KeyMapExact:
            Prio: Priority
          ValueMap:
            Priority:
              ValueMapExact:
                Critical: 5 Very High
                Information: 1 Very Low
                Warning: 3 Normal
          ValueMapDefault:
            MapTo: 3 Normal
            MapType: MapTo
        Type: Simple
      MappingOutbound:
        Config:
          KeyMapDefault:
            MapTo: ''
            MapType: Ignore
          KeyMapExact:
            Priority: Prio
          ValueMap:
            Prio:
              ValueMapExact:
                1 Very Low: Information
                3 Normal: Warning
                5 Very High: Critical
          ValueMapDefault:
            MapTo: ''
            MapType: Ignore
        Type: Simple
      Type: Test::Test
  Transport:
    Config:
      MaxLength: 10000000
      NameSpace: http://www.example.com/actions
    Type: HTTP::SOAP
RemoteSystem: remote.system.description.example.com
Requester:
  Invoker:
    Test:
      Description: This is a test invoker
      Events:
        - Asynchronous: 1
          Event: TicketCreate
        - Asynchronous: 0
          Event: ArticleUpdate
      MappingInbound:
        Type: Simple
      MappingOutbound:
        Type: Simple
      Type: Test::Test
  Transport:
    Config:
      Authentication:
        Password: '*******'
        Type: BasicAuth
        User: otrs
      Encoding: utf-8
      Endpoint: http://www.example.com:8080/endpoint
      NameSpace: http://www.example.com/actions
      SOAPAction: Yes
      SOAPActionSeparator: '#'
    Type: HTTP::SOAP
```

11.6.1. Описание Конфигурации

11.6.1.1. Общий

- Описание: краткий текст, описывающий веб-сервис.
- RemoteSystem: краткое описание Удаленной Системы.
- Debugger: хранилище для хранения настроек отладчика.
- Provider: хранилище для хранения настроек поставщика.

- Requester: хранилище для хранения настроек о запрашивающей стороне.

11.6.1.2. Отладчик

- DebugThreshold: уровень отладчика

  Возможные значения

  - debug: все журналы хранятся в базе данных.
  - info: информация, уведомления и журнал ошибкок хранятся в базе данных.
  - notice: ошибки уровня уведомление (notice) и ошибка (error) хранятся в базе данных.
  - error: только ошибки уровня error хранятся в базе данных.

11.6.1.3. Поставщик

- Operation: хранилище для настроек каждой операции.
- Transport: хранилище для настроек поставщика транспортой сети.

11.6.1.3.1. Операция

- <OperationName>: Уникальное название операции, хранилище для своих собственных настроек (от 0..n, но без повторений).

11.6.1.3.1.1. <OperationName>

Этот раздел основан на операция с типом "Test::Test" другие операции могут содержать больше или другие настройки.

- Описание: краткий текст, описывающий операцию.
- MappingInbound: хранилище для настроек отображаемых данных для данных из входящего запроса.
- MappingOutbound: хранилище для отображающихся настроек для исходящих данных.
- Тип бэкэнд операция в формате Controller::Operation.

11.6.1.3.1.1.1. MappingInbound

Этот раздел основан на отображаемых значениях с типом "Simple". Другие отображаемые значения могут содержать больше или другие настройки.

- Config: хранилище настроек для этих отображаемых значений.
- Тип: mapping backend.

11.6.1.3.1.1.1.1. Config

- KeyMapDefault: хранилище для всех не отображаемых значений ключей.
- ValueMapDefault: хранилище для всех не отображаемых значений настроек.
- KeyMapExact: хранилище для всех ключей точных отображаемых значений(от 0 .. 1).
- KeyMapRegEx: хранилище для отображаемых значений всех регулярных выражений.

- ValueMap: хранилище для всех значений отображений (mapping-а).

11.6.1.3.1.1.1.1.1. KeyMapDefault

- MapTo: новое значение, которое будет использоваться (применима только если параметр MapType установлен в значение MapTo).
- MapType: правило для отображения.

Возможные значения

- Keep: оставить без изменений.
- Ignore: понизить.
- MapTo: заменить на значение MapTo.

11.6.1.3.1.1.1.1.2. ValueMapDefault

Тоже что и KeyMapDefault.

11.6.1.3.1.1.1.1.3. KeyMapExact

- <oldkey>: <newkey> (от 0 .. n но без повторений).

11.6.1.3.1.1.1.1.4. KeyMapRegEx

- <oldkey(RegEx)>: <newkey> (от 0 .. n но без повторений).

11.6.1.3.1.1.1.1.5. ValueMap

- <newkey>: контейнер для отображаемых значений для этого нового ключа (зависит от новых ключей от KeyMapExact и KeyMapRegEx).

11.6.1.3.1.1.1.1.5.1. <newkey>

- ValueMapExact: хранилище для всех точных отображаемых значений (от 0 .. 1).
- ValueMapRegEx: хранилище для отображаемых значений всех регулярных выражений (от 0 .. 1).

11.6.1.3.1.1.1.1.5.1.1. valueMapExact

- <oldvalue>: <newvalue> (cardinality 0 .. n без повторений).

11.6.1.3.1.1.1.1.5.1.2. ValueMapRegEx

- <oldvalue(RegEx)>: <newvalue> (от 0 .. n без повторений).

11.6.1.3.1.1.2. MappingOutbound

Тоже что и MappingInbound.

11.6.1.3.1.1.3. Передача

Этот раздел основана на сетевом поставщике HTTP::SOAP, другие поставщики могут содержать больше или другие настройки.

- Config: хранилище для конкретных параметров конфигурации транспортной сети.
- Type: the provider network transport backend.

11.6.1.3.1.1.3.1. Config

- MaxLength: максимальная длина в байтах для чтения SOAP-сообщений системой OTRS.
- NameSpace: URI , который дает контекст для всех операций, принадлежащих этому веб-сервису.

11.6.1.4. Запрашиваемая сторона

- Invoker: хранилище для настроек каждой метки.
- Transport: хранилище для настроек транспортной сети запрашивающей стороны.

11.6.1.4.1. Запрашивающая сторона

- <InvokerName>: Уникальное имя для вызывающей стороны, хранилище для своих собственых настроек (от 0..n, без повторов).

11.6.1.4.1.1. <InvokerName>

Этот раздел основан на типе "Test::Test", другие invoker-ры могут иметь больше или другие настройки.

- Описание: короткий текст для описания вызывающей стороны
- События: хранилище настроек не именованного списка действий.
- MappingInbound: хранилище для настроек отображаемых значений для данных поступающих от ответов.
- MappingOutbound: хранилище для настроек отображаемых значений для исходящих данных запроса.
- Тип: invoker backend, в формате Controller::Invoker.

11.6.1.4.1.1.1. События

- *List Element*: (от 0 .. n)
  - Asynchronous: установить если вызов вызывающей стороны будет возложен на Планировщика

    Возможные значения

    - 0: не обрабатывается Планировщиком (Scheduler).
    - 1: Обрабатывается Планировщиком (Scheduler).
  - Event: имя события триггера.

    Возможные Значения (для событий заявки)

    - TicketCreate
    - TicketDelete
    - TicketTitleUpdate
    - TicketUnlockTimeoutUpdate

- TicketQueueUpdate
- TicketTypeUpdate
- TicketServiceUpdate
- TicketSLAUpdate
- TicketCustomerUpdate
- TicketFreeTextUpdate
- TicketFreeTimeUpdate
- TicketPendingTimeUpdate
- TicketLockUpdate
- TicketArchiveFlagUpdate
- TicketStateUpdate
- TicketOwnerUpdate
- TicketResponsibleUpdate
- TicketPriorityUpdate
- HistoryAdd
- HistoryDelete
- TicketAccountTime
- TicketMerge
- TicketSubscribe
- TicketUnsubscribe
- TicketFlagSet
- TicketFlagDelete
- TicketSlaveLinkAdd
- TicketSlaveLinkDelete
- TicketMasterLinkDelete

Возможные Значения (для действий над статьями)

- Действия над статьями
- ArticleCreate
- ArticleFreeTextUpdate

- ArticleUpdate
- ArticleSend
- ArticleBounce
- ArticleAgentNotification
- ArticleCustomerNotification
- ArticleAutoResponse
- ArticleFlagSet
- ArticleFlagDelete
- ArticleAgentNotification
- ArticleCustomerNotification

11.6.1.4.1.1.2. MappingInbound

То же что и Действие MappingInbound

11.6.1.4.1.1.3. MappingOutbound

Тоже что и Действие MappingInbound.

11.6.1.4.1.1.4. Передача

Этот раздел основан на транспортной сети запрашивающей стороны HTTP::SOAP, транспортные сети могут содержать больше или другие настройки.

- Config: хранилище для конкретных параметров конфигурации транспортной сети.
- Тип: бэкэнд транспортной сети запрашивающей стороны.

11.6.1.4.1.1.4.1. Config

- Authentication: a container for authentication settings.
- Encoding: кодирование сообщения SOAP-запроса
- Endpoint: URI Уаделнного Сервера веб-службы для приема запросов OTRS.
- NameSpace: URI который предоставлят связь для всех invoker-ов, которые принадлежат этому веб-серверу.
- SOAPAction: для отправки пустого или заполненого SOAPAction-заголовка в формате SOAP сообщения (in "<NameSpace> <Separator> <Action>" format).

  ### Возможные значения

  - YES: для отправки заполненого заголовка SOAPAction.
  - No: для отправки пустого обработчика SOAP-действия.
- SOAPActionSeparator: для установки <Separator> из заполненых SOAPAction заголовков.

Возможные значения

- '/': используется для .net веб-сервисов.
- '#': используется для всех веб-сервисов на базе протокола REST.

11.6.1.4.1.1.4.1.1. Authentication

- User: имя привилегированого пользователя, который имеет доступ к удаленному веб-сервису.
- Пароль: пароль для привелигированого пользователя в формате обычного текста.
- Тип: тип аутентификации.

11.7. Коннекторы (Connectors)

A Connector is in essence a set of actions that are either called Operations if OTRS acts as a web service provider or Invokers if OTRS acts as a web service requester. But it can also include special Mappings or Transports.

One Connector can either have only Operations, Only Invokers or both. A connector can even use parts of other connectors like the Mappings or Transports if they are not to specific for the Connector that is trying to implement them.

In other words a Connector is not limited to just the Controller layer but it can be extended to Data Mapping or Network Transport layers if needed.

Due to the modular design of the Generic Interface a Connector can be seen as a plug-in; this means that by adding Connectors the capabilities of the generic interface can be extended using: OTRS Feature add ons, OTRS Custom modules, 3rd Party modules, and so on.

11.7.1. Комплект Коннекторов (Connectors)

В комплекте с этой версией OTRS следующие коннекторы готовы к использованию.

- Session
- Заявк

11.7.1.1. Session Connector

Этот коннектор может создать действительный SessionID, который в дальнейшем может быть использован в любой другой операции.

Поставщики:

- Операции:
  - SessionCreate

11.7.1.1.1. Операции

11.7.1.1.1.1. SessionCreate

Creates a new new valid SessionID to be used in other operations from other connectors like TicketCreate.

Примечание

To use the SessionID in other operations from other connectors it is necessary that the operation implements authentication by SessionID. All the rest of the bundled operations are capable of accepting a valid SessionID as an authentication method.

Возможные Атрибуты:

```
<SessionCreate>
   <!--You have a MANDATORY CHOICE of the next 2 items at this level-->
   <!--Optional:-->
   <UserLogin>?</UserLogin>
   <!--Optional:-->
   <CustomerUserLogin>?</CustomerUserLogin>
   <!--Optional:-->
   <Password>?</Password>
</SessionCreate>
```

11.7.1.2. Коннектор Заявки

This connector supplies the basic functionality to interact with tickets

Поставщики:

- Операции:
  - TicketCreate
  - TicketUpdate
  - TicketGet
  - TicketSearch

11.7.1.2.1. Операции

11.7.1.2.1.1. TicketCreate

Provides an interface to create a ticket in OTRS. A ticket must contain an Article and can contain several attachments, all defined Dynamic Fields can be also set on TicketCreate operation.

Возможные Атрибуты:

```
<TicketCreate>
   <!--You have a MANDATORY CHOICE of the next 3 items at this level-->
   <!--Optional:-->
   <UserLogin>?</UserLogin>
   <!--Optional:-->
   <CustomerUserLogin>?</CustomerUserLogin>
   <!--Optional:-->
   <SessionID>?</SessionID>
   <!--Optional:-->
   <Password>?</Password>
   <Ticket>
      <Title>?</Title>
      <!--You have a MANDATORY CHOICE of the next 2 items at this level-->
      <!--Optional:-->
```

```
            <QueueID>?</QueueID>
            <!--Optional:-->
            <Queue>?</Queue>
            <!--You have a CHOICE of the next 2 items at this level-->
            <!--Optional:-->
            <TypeID>?</TypeID>
            <!--Optional:-->
            <Type>?</Type>
            <!--You have a CHOICE of the next 2 items at this level-->
            <!--Optional:-->
            <ServiceID>?</ServiceID>
            <!--Optional:-->
            <Service>?</Service>
            <!--You have a CHOICE of the next 2 items at this level-->
            <!--Optional:-->
            <SLAID>?</SLAID>
            <!--Optional:-->
            <SLA>?</SLA>
            <!--You have a MANDATORY CHOICE of the next 2 items at this level-->
            <!--Optional:-->
            <StateID>?</StateID>
            <!--Optional:-->
            <State>?</State>
            <!--You have a MANDATORY CHOICE of the next 2 items at this level-->
            <!--Optional:-->
            <PriorityID>?</PriorityID>
            <!--Optional:-->
            <Priority>?</Priority>
            <!--You have a CHOICE of the next 2 items at this level-->
            <!--Optional:-->
            <OwnerID>?</OwnerID>
            <!--Optional:-->
            <Owner>?</Owner>
            <!--You have a CHOICE of the next 2 items at this level-->
            <!--Optional:-->
            <ResponsibleID>?</ResponsibleID>
            <!--Optional:-->
            <Responsible>?</Responsible>
            <CustomerUser>?</CustomerUser>
            <!--Optional:-->
            <CustomerID>?</CustomerID>
            <!--Optional:-->
            <PendingTime>
               <Year>?</Year>
               <Month>?</Month>
               <Day>?</Day>
               <Hour>?</Hour>
               <Minute>?</Minute>
            </PendingTime>
         </Ticket>
         <Article>
            <!--You have a CHOICE of the next 2 items at this level-->
            <!--Optional:-->
            <ArticleTypeID>?</ArticleTypeID>
            <!--Optional:-->
            <ArticleType>?</ArticleType>
            <!--You have a CHOICE of the next 2 items at this level-->
            <!--Optional:-->
            <SenderTypeID>?</SenderTypeID>
            <!--Optional:-->
            <SenderType>?</SenderType>
            <!--Optional:-->
            <From>?</From>
            <Subject>?</Subject>
            <Body>?</Body>
            <!--You have a CHOICE of the next 2 items at this level-->
            <!--Optional:-->
            <ContentType>?</ContentType>
            <Charset>?</Charset>
            <MimeType>?</MimeType>
            <!--Optional:-->
            <HistoryType>?</HistoryType>
            <!--Optional:-->
            <HistoryComment>?</HistoryComment>
            <!--Optional:-->
            <AutoResponseType>?</AutoResponseType>
            <!--Optional:-->
            <TimeUnit>?</TimeUnit>
            <!--Optional:-->
            <NoAgentNotify>?</NoAgentNotify>
            <!--Zero or more repetitions:-->
            <ForceNotificationToUserID>?</ForceNotificationToUserID>
            <!--Zero or more repetitions:-->
            <ExcludeNotificationToUserID>?</ExcludeNotificationToUserID>
            <!--Zero or more repetitions:-->
            <ExcludeMuteNotificationToUserID>?</ExcludeMuteNotificationToUserID>
         </Article>
         <!--Zero or more repetitions:-->
         <DynamicField>
            <Name>?</Name>
            <!--1 or more repetitions:-->
            <Value>?</Value>
         </DynamicField>
         <!--Zero or more repetitions:-->
         <Attachment>
            <Content>cid:61886944659</Content>
            <ContentType>?</ContentType>
            <Filename>?</Filename>
         </Attachment>
      </TicketCreate>
```

11.7.1.2.1.2. TicketUpdate

TicketUpdate operation adds the capability to modify attributes from an existing ticket or to add a new article, including attachments and all defined dynamic fields for the ticket and the new article.

Примечание

It is not necessary to create a new article to modify a ticket attribute.

Возможные Атрибуты:

```
   <TicketUpdate>
      <!--You have a MANDATORY CHOICE of the next 3 items at this level-->
      <!--Optional:-->
      <UserLogin>?</UserLogin>
      <!--Optional:-->
      <CustomerUserLogin>?</CustomerUserLogin>
      <!--Optional:-->
      <SessionID>?</SessionID>
      <!--Optional:-->
      <Password>?</Password>
      <!--You have a CHOICE of the next 2 items at this level-->
      <TicketID>?</TicketID>
      <TicketNumber>?</TicketNumber>
      <!--Optional:-->
      <Ticket>
         <!--Optional:-->
```

```
           <Title>?</Title>
           <!--You have a CHOICE of the next 2 items at this level-->
           <!--Optional:-->
           <QueueID>?</QueueID>
           <!--Optional:-->
           <Queue>?</Queue>
           <!--You have a CHOICE of the next 2 items at this level-->
           <!--Optional:-->
           <TypeID>?</TypeID>
           <!--Optional:-->
           <Type>?</Type>
           <!--You have a CHOICE of the next 2 items at this level-->
           <!--Optional:-->
           <ServiceID>?</ServiceID>
           <!--Optional:-->
           <Service>?</Service>
           <!--You have a CHOICE of the next 2 items at this level-->
           <!--Optional:-->
           <SLAID>?</SLAID>
           <!--Optional:-->
           <SLA>?</SLA>
           <!--You have a CHOICE of the next 2 items at this level-->
           <!--Optional:-->
           <StateID>?</StateID>
           <!--Optional:-->
           <State>?</State>
           <!--You have a CHOICE of the next 2 items at this level-->
           <!--Optional:-->
           <PriorityID>?</PriorityID>
           <!--Optional:-->
           <Priority>?</Priority>
           <!--You have a CHOICE of the next 2 items at this level-->
           <!--Optional:-->
           <OwnerID>?</OwnerID>
           <!--Optional:-->
           <Owner>?</Owner>
           <!--You have a CHOICE of the next 2 items at this level-->
           <!--Optional:-->
           <ResponsibleID>?</ResponsibleID>
           <!--Optional:-->
           <Responsible>?</Responsible>
           <!--Optional:-->
           <CustomerUser>?</CustomerUser>
           <!--Optional:-->
           <CustomerID>?</CustomerID>
           <!--Optional:-->
           <PendingTime>
              <Year>?</Year>
              <Month>?</Month>
              <Day>?</Day>
              <Hour>?</Hour>
              <Minute>?</Minute>
           </PendingTime>
        </Ticket>
        <!--Optional:-->
        <Article>
           <!--You have a CHOICE of the next 2 items at this level-->
           <!--Optional:-->
           <ArticleTypeID>?</ArticleTypeID>
           <!--Optional:-->
           <ArticleType>?</ArticleType>
           <!--You have a CHOICE of the next 2 items at this level-->
           <!--Optional:-->
           <SenderTypeID>?</SenderTypeID>
           <!--Optional:-->
           <SenderType>?</SenderType>
           <!--Optional:-->
```

```
      <From>?</From>
      <Subject>?</Subject>
      <Body>?</Body>
      <!--You have a CHOICE of the next 2 items at this level-->
      <!--Optional:-->
      <ContentType>?</ContentType>
      <Charset>?</Charset>
      <MimeType>?</MimeType>
      <!--Optional:-->
      <HistoryType>?</HistoryType>
      <!--Optional:-->
      <HistoryComment>?</HistoryComment>
      <!--Optional:-->
      <AutoResponseType>?</AutoResponseType>
      <!--Optional:-->
      <TimeUnit>?</TimeUnit>
      <!--Optional:-->
      <NoAgentNotify>?</NoAgentNotify>
      <!--Zero or more repetitions:-->
      <ForceNotificationToUserID>?</ForceNotificationToUserID>
      <!--Zero or more repetitions:-->
      <ExcludeNotificationToUserID>?</ExcludeNotificationToUserID>
      <!--Zero or more repetitions:-->
      <ExcludeMuteNotificationToUserID>?</ExcludeMuteNotificationToUserID>
   </Article>
   <!--Zero or more repetitions:-->
   <DynamicField>
      <Name>?</Name>
      <!--1 or more repetitions:-->
      <Value>?</Value>
   </DynamicField>
   <!--Zero or more repetitions:-->
   <Attachment>
      <Content>cid:166861569966</Content>
      <ContentType>?</ContentType>
      <Filename>?</Filename>
   </Attachment>
</TicketUpdate>
```

11.7.1.2.1.3. TicketGet

This operation is used to get all the attributes of a ticket including the dynamic fields, all articles and all of the attachments that belong to the ticket.

Возможные Атрибуты:

```
<TicketGet>
   <!--You have a MANDATORY CHOICE of the next 3 items at this level-->
   <!--Optional:-->
   <UserLogin>?</UserLogin>
   <!--Optional:-->
   <CustomerUserLogin>?</CustomerUserLogin>
   <!--Optional:-->
   <SessionID>?</SessionID>
   <!--Optional:-->
   <Password>?</Password>
   <!--1 or more repetitions:-->
   <TicketID>?</TicketID>
   <!--Optional:-->
   <DynamicFields>?</DynamicFields>
   <!--Optional:-->
   <Extended>?</Extended>
   <!--Optional:-->
   <AllArticles>?</AllArticles>
   <!--Optional:-->
   <ArticleSenderType>?</ArticleSenderType>
   <!--Optional:-->
   <ArticleOrder>?</ArticleOrder>
   <!--Optional:-->
   <ArticleLimit>?</ArticleLimit>
   <!--Optional:-->
   <Attachments>?</Attachments>
</TicketGet>
```

11.7.1.2.1.4. TicketSearch

TicketSearch operation returns a list of Ticket IDs that matches a predefined criteria.

Возможные Атрибуты:

```
<TicketSearch>
   <!--You have a MANDATORY CHOICE of the next 3 items at this level-->
   <!--Optional:-->
   <UserLogin>?</UserLogin>
   <!--Optional:-->
   <CustomerUserLogin>?</CustomerUserLogin>
   <!--Optional:-->
   <SessionID>?</SessionID>
   <!--Optional:-->
   <Password>?</Password>
   <!--Optional:-->
   <Limit>?</Limit>
   <!--Zero or more repetitions:-->
   <TicketNumber>?</TicketNumber>
   <!--Zero or more repetitions:-->
   <Title>?</Title>
   <!--Zero or more repetitions:-->
   <Queues>?</Queues>
   <!--Zero or more repetitions:-->
   <QueueIDs>?</QueueIDs>
   <!--Optional:-->
   <UseSubQueues>?</UseSubQueues>
   <!--Zero or more repetitions:-->
   <Types>?</Types>
   <!--Zero or more repetitions:-->
   <TypeIDs>?</TypeIDs>
   <!--Zero or more repetitions:-->
   <States>?</States>
   <!--Zero or more repetitions:-->
   <StateIDs>?</StateIDs>
   <!--Zero or more repetitions:-->
   <StateType>?</StateType>
   <!--Zero or more repetitions:-->
   <StateTypeIDs>?</StateTypeIDs>
   <!--Zero or more repetitions:-->
   <Priorities>?</Priorities>
   <!--Zero or more repetitions:-->
   <PriorityIDs>?</PriorityIDs>
   <!--Zero or more repetitions:-->
   <Services>?</Services>
   <!--Zero or more repetitions:-->
   <ServiceIDs>?</ServiceIDs>
   <!--Zero or more repetitions:-->
   <SLAs>?</SLAs>
   <!--Zero or more repetitions:-->
   <SLAIDs>?</SLAIDs>
```

```
         <!--Zero or more repetitions:-->
         <Locks>?</Locks>
         <!--Zero or more repetitions:-->
         <LockIDs>?</LockIDs>
         <!--Zero or more repetitions:-->
         <OwnerIDs>?</OwnerIDs>
         <!--Zero or more repetitions:-->
         <ResponsibleIDs>?</ResponsibleIDs>
         <!--Zero or more repetitions:-->
         <WatchUserIDs>?</WatchUserIDs>
         <!--Zero or more repetitions:-->
         <CustomerID>?</CustomerID>
         <!--Zero or more repetitions:-->
         <CustomerUserLogin>?</CustomerUserLogin>
         <!--Zero or more repetitions:-->
         <CreatedUserIDs>?</CreatedUserIDs>
         <!--Zero or more repetitions:-->
         <CreatedTypes>?</CreatedTypes>
         <!--Zero or more repetitions:-->
         <CreatedTypeIDs>?</CreatedTypeIDs>
         <!--Zero or more repetitions:-->
         <CreatedPriorities>?</CreatedPriorities>
         <!--Zero or more repetitions:-->
         <CreatedPriorityIDs>?</CreatedPriorityIDs>
         <!--Zero or more repetitions:-->
         <CreatedStates>?</CreatedStates>
         <!--Zero or more repetitions:-->
         <CreatedStateIDs>?</CreatedStateIDs>
         <!--Zero or more repetitions:-->
         <CreatedQueues>?</CreatedQueues>
         <!--Zero or more repetitions:-->
         <CreatedQueueIDs>?</CreatedQueueIDs>
         <!--Zero or more repetitions:-->
         <DynamicFields>
            <!--You have a MANDATORY CHOICE of the next 6 items at this level-->
            <!--Optional:-->
            <Equals>?</Equals>
            <!--Optional:-->
            <Like>?</Like>
            <!--Optional:-->
            <GreaterThan>?</GreaterThan>
            <!--Optional:-->
            <GreaterThanEquals>?</GreaterThanEquals>
            <!--Optional:-->
            <SmallerThan>?</SmallerThan>
            <!--Optional:-->
            <SmallerThanEquals>?</SmallerThanEquals>
         </DynamicFields>
         <!--Optional:-->
         <Ticketflag>
            <!--Optional:-->
            <Seen>?</Seen>
         </Ticketflag>
         <!--Optional:-->
         <From>?</From>
         <!--Optional:-->
         <To>?</To>
         <!--Optional:-->
         <Cc>?</Cc>
         <!--Optional:-->
         <Subject>?</Subject>
         <!--Optional:-->
         <Body>?</Body>
         <!--Optional:-->
         <FullTextIndex>?</FullTextIndex>
         <!--Optional:-->
         <ContentSearch>?</ContentSearch>
         <!--Optional:-->
         <ConditionInline>?</ConditionInline>
         <!--Optional:-->
         <ArticleCreateTimeOlderMinutes>?</ArticleCreateTimeOlderMinutes>
         <!--Optional:-->
         <ArticleCreateTimeNewerMinutes>?</ArticleCreateTimeNewerMinutes>
         <!--Optional:-->
         <ArticleCreateTimeNewerDate>?</ArticleCreateTimeNewerDate>
         <!--Optional:-->
         <ArticleCreateTimeOlderDate>?</ArticleCreateTimeOlderDate>
         <!--Optional:-->
         <TicketCreateTimeOlderMinutes>?</TicketCreateTimeOlderMinutes>
         <!--Optional:-->
         <ATicketCreateTimeNewerMinutes>?</ATicketCreateTimeNewerMinutes>
         <!--Optional:-->
         <TicketCreateTimeNewerDate>?</TicketCreateTimeNewerDate>
         <!--Optional:-->
         <TicketCreateTimeOlderDate>?</TicketCreateTimeOlderDate>
         <!--Optional:-->
         <TicketChangeTimeOlderMinutes>?</TicketChangeTimeOlderMinutes>
         <!--Optional:-->
         <TicketChangeTimeNewerMinutes>?</TicketChangeTimeNewerMinutes>
         <!--Optional:-->
         <TicketChangeTimeNewerDate>?</TicketChangeTimeNewerDate>
         <!--Optional:-->
         <TicketChangeTimeOlderDate>?</TicketChangeTimeOlderDate>
         <!--Optional:-->
         <TicketCloseTimeOlderMinutes>?</TicketCloseTimeOlderMinutes>
         <!--Optional:-->
         <TicketCloseTimeNewerMinutes>?</TicketCloseTimeNewerMinutes>
         <!--Optional:-->
         <TicketCloseTimeNewerDate>?</TicketCloseTimeNewerDate>
         <!--Optional:-->
         <TicketCloseTimeOlderDate>?</TicketCloseTimeOlderDate>
         <!--Optional:-->
         <TicketPendingTimeOlderMinutes>?</TicketPendingTimeOlderMinutes>
         <!--Optional:-->
         <TicketPendingTimeNewerMinutes>?</TicketPendingTimeNewerMinutes>
         <!--Optional:-->
         <TicketPendingTimeNewerDate>?</TicketPendingTimeNewerDate>
         <!--Optional:-->
         <TicketPendingTimeOlderDate>?</TicketPendingTimeOlderDate>
         <!--Optional:-->
         <TicketEscalationTimeOlderMinutes>?</TicketEscalationTimeOlderMinutes>
         <!--Optional:-->
         <TTicketEscalationTimeNewerMinutes>?</TTicketEscalationTimeNewerMinutes>
         <!--Optional:-->
         <TicketEscalationTimeNewerDate>?</TicketEscalationTimeNewerDate>
         <!--Optional:-->
         <TicketEscalationTimeOlderDate>?</TicketEscalationTimeOlderDate>
         <!--Optional:-->
         <ArchiveFlags>?</ArchiveFlags>
         <!--Zero or more repetitions:-->
         <OrderBy>?</OrderBy>
         <!--Zero or more repetitions:-->
         <SortBy>?</SortBy>
         <!--Zero or more repetitions:-->
         <CustomerUserID>?</CustomerUserID>
      </TicketSearch>
```

11.7.2. Примеры:

11.7.2.1. Конфигурация Веб-сервиса

The following is a basic but complete web service configuration file in YAML format to use all the Ticket Connector operations. In order to use it in OTRS you need to copy the content, save it into a file called GenericTicketConnector.yml, and import it into OTRS in the Web Services screen in the Admin panel by clicking in the "Add web service" action from the overview screen and then clicking in the "Import web service" action in the add screen.

```
---
Debugger:
  DebugThreshold: debug
  TestMode: 0
Description: Ticket Connector Sample
FrameworkVersion: 3.2.x CVS
Provider:
  Operation:
    SessionCreate:
      Description: Creates a Session
      MappingInbound: {}
      MappingOutbound: {}
      Type: Session::SessionCreate
    TicketCreate:
      Description: Creates a Ticket
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketCreate
    TicketUpdate:
      Description: Updates a Ticket
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketUpdate
    TicketGet:
      Description: Retrieve Ticket data
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketGet
    TicketSearch:
      Description: Search for Tickets
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketSearch
  Transport:
    Config:
      MaxLength: 100000000
      NameSpace: http://www.otrs.org/TicketConnector/
    Type: HTTP::SOAP
RemoteSystem: ''
Requester:
  Transport:
    Type: ''
```

11.7.2.2. Запрашивающая сторона Perl SOAP

The following code is a Perl script that can connect to OTRS via the generic interface. In order to perform the operations provided by the Ticket Connector, it uses two Perl CPAN modules SOAP::Lite and Data::Dumper. Please make sure that your environment is capable to use these modules before you try to run the scipt.

```
#!/usr/bin/perl -w
# --
# otrs.SOAPRequest.pl - sample to send a SOAP request to OTRS Generic Interface Ticket
 Connector
# Copyright (C) 2001-2014 OTRS AG, http://otrs.com/
# --
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU AFFERO General Public License as published by
# the Free Software Foundation; either version 3 of the License, or
# any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU Affero General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
# or see http://www.gnu.org/licenses/agpl.txt.
# --

use strict;
use warnings;

# use ../ as lib location
use File::Basename;
use FindBin qw($RealBin);
use lib dirname($RealBin);

use SOAP::Lite;
use Data::Dumper;

# ---
# Variables to be defined.

# this is the URL for the web service
# the format is
# <HTTP_TYPE>:://<OTRS_FQDN>/nph-genericinterface.pl/Webservice/<WEB_SERVICE_NAME>
# or
# <HTTP_TYPE>:://<OTRS_FQDN>/nph-genericinterface.pl/WebserviceID/<WEB_SERVICE_ID>
my $URL = 'http://localhost/otrs/nph-genericinterface.pl/Webservice/GenericTicketConnector';

# this name space should match the specified name space in the SOAP transport for the web
 service.
my $NameSpace = 'http://www.otrs.org/TicketConnector/';

# this is operation to execute, it could be TicketCreate, TicketUpdate, TicketGet, TicketSearch
# or SessionCreate. and they must to be defined in the web service.
my $Operation = 'TicketCreate';

# this variable is used to store all the parameters to be included on a request in XML format.
 Each
# operation has a determined set of mandatory and non mandatory parameters to work correctly.
 Please
# check the OTRS Admin Manual in order to get a complete list of parameters.
my $XMLData = '
<UserLogin>some user login</UserLogin>
<Password>some password</Password>
<Ticket>
    <Title>some title</Title>
    <CustomerUser>some customer user login</CustomerUser>
    <Queue>some queue</Queue>
    <State>some state</State>
    <Priority>some priority</Priority>
</Ticket>
```

```
<Article>
    <Subject>some subject</Subject>
    <Body>some body</Body>
    <ContentType>text/plain; charset=utf8</ContentType>
</Article>
';

# ---

# create a SOAP::Lite data structure from the provided XML data structure.
my $SOAPData = SOAP::Data
    ->type( 'xml' => $XMLData );

my $SOAPObject = SOAP::Lite
    ->uri($NameSpace)
    ->proxy($URL)
    ->$Operation($SOAPData);

# check for a fault in the soap code.
if ( $SOAPObject->fault ) {
    print $SOAPObject->faultcode, " ", $SOAPObject->faultstring, "\n";
}

# otherwise print the results.
else {

    # get the XML response part from the SOAP message.
    my $XMLResponse = $SOAPObject->context()->transport()->proxy()->http_response()->content();

    # deserialize response (convert it into a perl structure).
    my $Deserialized = eval {
        SOAP::Deserializer->deserialize($XMLResponse);
    };

    # remove all the headers and other not needed parts of the SOAP message.
    my $Body = $Deserialized->body();

    # just output relevant data and no the operation name key (like TicketCreateResponse).
    for my $ResponseKey ( keys %{$Body} ) {
        print Dumper( $Body->{$ResponseKey} );
    }
}
```

12. Планировщик OTRS

The OTRS Scheduler is an independent system process that executes tasks in background. These kind of processes are know as *daemons* in Unix / Linux systems or as *services* on Windows environments. It is independent but that doesn't mean that the Scheduler does everything alone, it is fully integrated into OTRS and can use any OTRS module as needed to complete each task.

For sanity reasons the Scheduler process needs to be restarted from time to time. This is done automatically by the scheduler process itself once a day, but it can be adjusted as needed using the SysConfig by editing the "Scheduler::RestartAfterSeconds" setting.

The OTRS Scheduler is a fully automated process, the only needed human interaction is to check its status periodically and start or stop it as needed.

Примечание

If the Scheduler is stopped for any reason, all pending tasks and new tasks registered when the Scheduler is stopped will be executed as soon as the Scheduler starts again (unless the tasks are set to be executed in the future).

12.1. Графический интерфейс планировщика

Планировщик не виден в Графическом Пользовательском Интерфейсе OTRS до того времени пока он остается работать.

12.1.1. Планировщик Не Запускает Уведомления

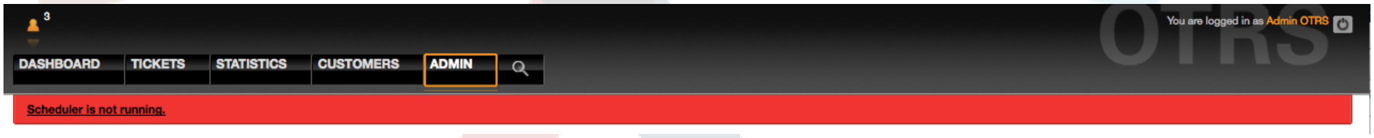
There are two different types of notifications if the system detects that the scheduler is not running. This detection is based on the update frequency of the Scheduler process. If the difference between current time and the last process update time is 2 times the process update frequency a warning message will be displayed in the OTRS notification area. If it is over 4 times the process frequency then an alert will be displayed instead.

Параметр Планировщика обновление времени может быть настроен с помощью параметра "Scheduler::PIDUpdateTime" в файле SysConfig.

If you see a warning message it is not always necessary to take an action, but it is highly recommended to check if the scheduler process is running. If you see an alert, then there is a high chance that the scheduler is in fact not running and should be started.

По умолчанию Планировщик не запускает уведомлений, если есть действительный веб-сервис, зарегистрированный в базе данных и он отображается только для пользователей из группы "admin".

Чтобы отключить уведомления (не рекомендуется) или же изменить или добавить группы уведомлений, пожалуйста, отредактируйте настройку "Frontend::NotifyModule###800-Scheduler-Check" в файле SysConfig.



Scheduler is not running.

Figure: Уведомления планировщика.

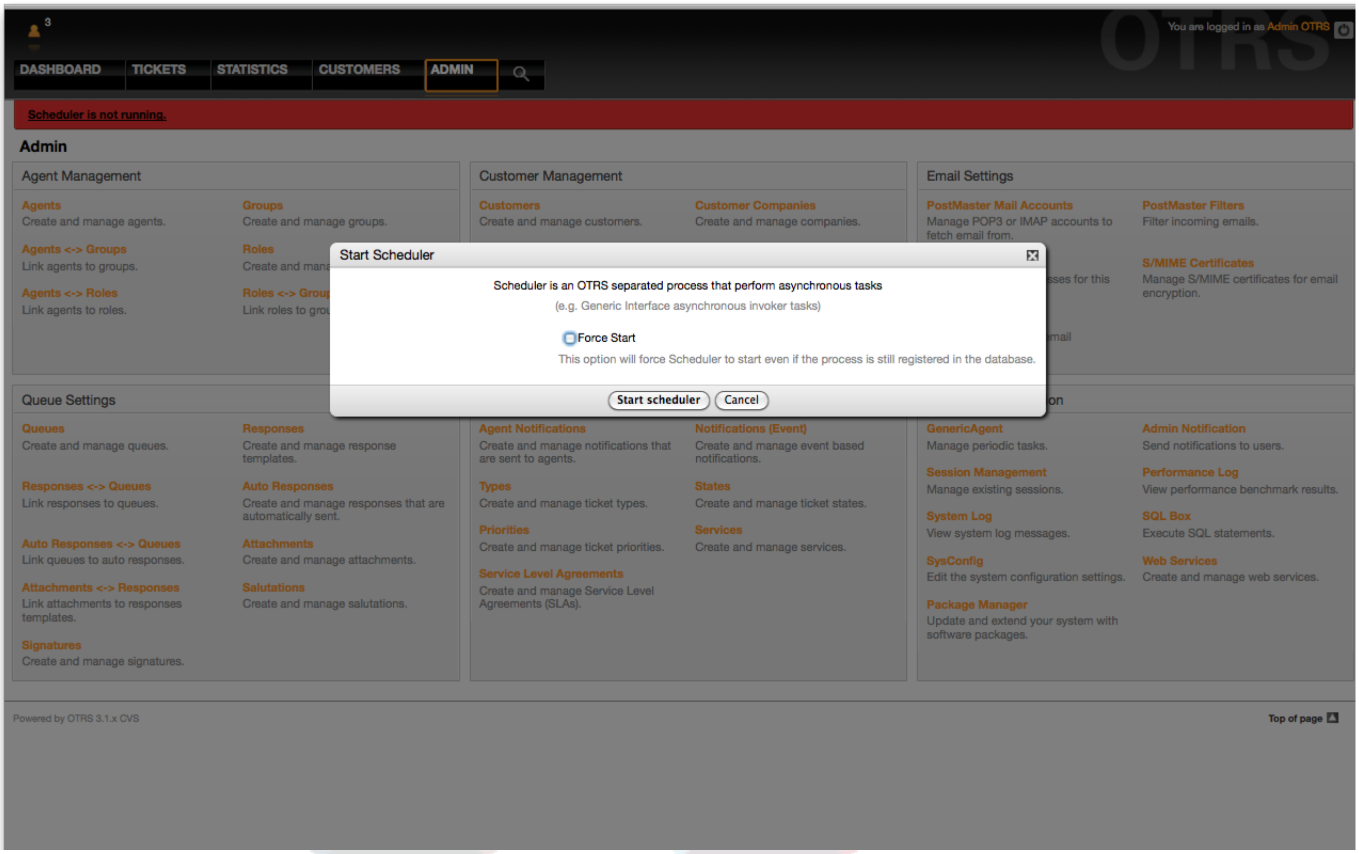
12.1.2. Запуск Планировщика

By clicking on the Scheduler not running notification link (either warning or alert) a dialog box will open to let you start the Scheduler process again. The Scheduler can be started normally or forced to start, by clicking on the appropriate check box in the dialog.

Примечание

A forced Scheduler start is only necessary if previous Scheduler process was terminated abnormally and the Process ID is still registered in the database.

Используйте инструменты командной строки, описанные ниже, чтобы иметь полный контроль над процессами Планировщика и иметь возможность проверить его реальное состояние.


DASHBOARD
TICKETS
STATISTICS
CUSTOMERS
ADMIN
You are logged in as Admin OTRS
Admin
Agent Management
Agents
Create and manage agents.
Groups
Create and manage groups.
Agents <-> Groups
Link agents to groups.
Roles
Agents <-> Roles
Link agents to roles.
Customer Management
Customers
Create and manage customers.
Customer Companies
Create and manage companies.
Email Settings
PostMaster Mail Accounts
Manage POP3 or IMAP accounts to fetch email from.
PostMaster Filters
Filter incoming emails.
S/MIME Certificates
Manage S/MIME certificates for email encryption.
Queue Settings
Queues
Create and manage queues.
Responses
Create and manage response templates.
Responses <-> Queues
Link responses to queues.
Auto Responses
Create and manage responses that are automatically sent.
Auto Responses <-> Queues
Link queues to auto responses.
Attachments
Create and manage attachments.
Attachments <-> Responses
Link attachments to responses templates.
Salutations
Create and manage salutations.
Signatures
Create and manage signatures.
Agent Notifications
Create and manage notifications that are sent to agents.
Notifications (Event)
Create and manage event based notifications.
Types
Create and manage ticket types.
States
Create and manage ticket states.
Priorities
Create and manage ticket priorities.
Services
Create and manage services.
Service Level Agreements
Create and manage Service Level Agreements (SLAs).
GenericAgent
Manage periodic tasks.
Admin Notification
Send notifications to users.
Session Management
Manage existing sessions.
Performance Log
View performance benchmark results.
System Log
View system log messages.
SQL Box
Execute SQL statements.
SysConfig
Edit the system configuration settings.
Web Services
Create and manage web services.
Package Manager
Update and extend your system with software packages.
Powered by OTRS 3.1.x CVS
Top of page
Start scheduler
Cancel

Scheduler is not running.

Start Scheduler

Scheduler is an OTRS separated process that perform asynchronous tasks

(e.g. Generic Interface asynchronous invoker tasks)

□ Force Start

This option will force Scheduler to start even if the process is still registered in the database.

Рисунок: Запуск Планировщика.

12.2. Интерфейс командной строки Планировщика

Командная строка Планировщика позволяет управлять процессами (Запуск / Остановка) или запрашивать их состояние. Также есть инструменты для регистрации процесса, который будет управлять операционной системой.

Included with OTRS there are two sets of CLI tools, one for Unix / Linux and another for Microsoft Windows.

12.2.1. Unix / Linux

12.2.1.1. Scheduler init.d files

Init.d files are special scripts that are called by the operating system at startup and shutdown.

OTRS provides init.d scripts to start / stop the OTRS Scheduler process automatically by the operating system. These scripts are located under OTRS_HOME/scripts.

The init.d scripts need to be copied to the correct location for your operating system. They need to have the proper permissions and some internal variables need to be set to work properly. If you used the OTRS RPMs to set up the system, this is taken care of automatically.

Init.d Скрипт Внутренних Переменных

- **OTRS_HOME** - путь установки OTRS.
- **User** - имя пользователя процесса apache.

- **Group** - apache обрабатывает группу пользователей с таким именем.

Примечание

Currently OTRS provides only init.d scripts for Linux platforms.

Таблица 4.7. List of init scripts and supported Operating Systems

Сценарий инициализации	Поддерживаемы операционные системы
otrs-scheduler-linux	Red Hat, Fedora, CentOS, SUSE, openSUSE, Debian, Ubuntu
otrs-scheduler-gentoo-init.d, otrs-scheduler-gentoo-conf.d	Gentoo

Пример 4.29. Example to start the OTRS Scheduler on linux

```
shell> /etc/init.d/otrs-scheduler-linux start
```

Возможные действия

- **start** запуск процесса OTRS-Планировщика.
- **stop** остановка процесса OTRS-Планировщика.
- **restart** перезапуск процесса OTRS-Планировщика.
- **status** запросить статус процесса OTRS-Планировщика.

Для корректной работы Планировщика необходима база данных, поскольку он регистрирует свой ID процесса, именно по этой причине необходимо:

- Выполните скрипт init.d для *запуска* процесса Планировщика, после того как запущен процес базы данных.
- Execute the Scheduler init.d script to *stop* the Scheduler before the database process shuts down.

Примечание

If you want the Scheduler to run at system startup, please read the documentation of the operating system to find out the right location to place the init.d scripts, how to configure them to run automatically and how to set the run order.

12.2.1.2. Файл "Демона" Планировщика

Это часть Планировщика, который остается работать в фоновом режиме, проверяя задания, которые нужно выполнить. Она также обеспечивает основные функции для управления процессом.

Все Unix / Linux системы используют файл **OTRS_HOME/bin/otrs.Scheduler.pl**.

Пример 4.30. Пример для Запуска OTRS-Планировщика

```
                    shell> OTRS_HOME/bin/otrs.Scheduler.pl -a start
```

Доступные Опции

- **-a** действие.

  #### Возможные значения

  - **start**- старт процесса Планировщика.
  - **stop**- остановка процесса Планировщика.
  - **status**- для запроса статуса процесса Планировщика.

- **-f** принудительный старт или остановка процесса Планировщика.

Пример 4.31. Example to force stop the OTRS Scheduler

```
                    shell> OTRS_HOME/bin/otrs.Scheduler.pl -a stop -f 1
```

Примечание

Force stop the Scheduler is used remove the process ID from the database when the scheduler is not running and the process is still registered.

Force start the Scheduler is used to start the Scheduler process if the scheduler is not running and the process is registered.

Force start or stop are only necessary if the start of the process is needed to be done before the process update time expires. Otherwise an expired entry in the database is discarded by normal start.

12.2.2. Windows

12.2.2.1. Установщик Планировщика

On Windows, the OTRS Scheduler is registered as a Service via the Windows Service Control Manager (SCM). In order to have the OTRS Scheduler process to be controlled by the SCM is necessary to register this service. If you used the Windows Installer this is taken care of by the installer, and there are no manual actions required.

OTRS предоставляет сценарий **OTRS_HOME/bin/otrs.Scheduler4WinInstaller.pl** для регистрации и отмены регистрации OTRS-планировщика в SCM.

Примечание

Registering and de-registering can only be performed by members of the Administrator group on Windows. You might need to right-click on cmd.exe and select 'Run as Administrator' before you can register the service correctly.

Пример 4.32. Example to register the OTRS Scheduler as a Windows Service

```
shell> perl OTRS_HOME/bin/otrs.Scheduler4WinInstaller.pl -a install
```

Доступные Опции

- **-a** действие.

Возможные значения

- **install**- для установки процесса Планировщика в Windows SCM.
- **remove**- для удаления процесса Планировщика из Windows SCM.

After installing into the Windows SCM the OTRS Scheduler process can be used like any other service on Windows. It can be started, stopped and restarted and can be configured to be started manually or automatic.

Примечание

Обратитесь к документации Windows для более подробного изучения Сервисов Windows и Windows SCM.

12.2.2.2. Сервисный Файл Планировщика

Это часть Планировщика, который остается работать в фоновом режиме, проверяя задания, которые нужно выполнить. Она также обеспечивает основные функции для управления процессом.

Microsoft Windows uses the file **OTRS_HOME/bin/otrs.Scheduler4Win.pl**.

Пример 4.33. Example to start the OTRS Scheduler on Windows

```
shell> OTRS_HOME/bin/otrs.Scheduler4Win.pl -a start
```

Доступные Опции

- **-a** действие.

Возможные значения

- **start**- старт процесса Планировщика.
- **stop**- остановка процесса Планировщика.
- **status**- для запроса статуса процесса Планировщика.

- **-f** принудительный старт или остановка процесса Планировщика.

Пример 4.34. Example to force stop the OTRS Scheduler on Windows

```
shell> OTRS_HOME/bin/otrs.Scheduler4Win.pl -a stop -f 1
```

Примечание

Force stopping the Scheduler is used to remove the process ID from the database when the scheduler is not running and the process is still registered.

Force starting the Scheduler is used to start the Scheduler process if the scheduler is not running and the process is still registered.

Force start or stop are only necessary if starting the process is needed to be done before the process update time expires. Otherwise an expired entry in the database would be discarded by a normal start.

Глава 5. Настройка

1. Списки Контроля Доступа (ACLs)

1.1. Введение

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 to supplement the existing permission system of roles and groups. Using ACLs, rudimental workflows within the system can be mapped, based on ticket attributes.

As yet, ACLs cannot be created using the SysConfig interface. They must be directly entered into the `Kernel/Config.pm` file. This chapter has some ACL examples which will walk you trough the process of defining ACL definitions, and a reference of all possible important ACL settings.

1.2. Примеры

Пример 5.1. Списки прав доступа (ACL) позволяют перемещать в очереди только заявки с приоритетом 5.

Этот пример показывает основную струкруру ACL. Для начала, она должна иметь название. В нашем случае это "ACL-Name-2". Обратите внимание, что перед запуском ACLs будут отсортированы по номерам, поэтому имена нужно использовать тщательно.

Secondly, you have a "Properties" section which is a filter for your tickets. All the criteria defined here will be applied to a ticket to determine if the ACL must be applied or not. In our example, a ticket will match if it is in the queue "Raw" and has priority "5 very high". This is also affected by changes in the form (e.g. if the ticket is the queue "raw" and had a priority "3 normal", but then priority drop-down is selected and the priority is changed now to "5 very high" will also match).

Наконец, раздел "Возможные" определяет изменения экрана. В нашем случае из имеющихся очередей только очередь "Alert" может быть выбрана на странице Заявки.

```
# ticket acl
$Self->{TicketAcl}->{'100-Example-ACL'} = {
    # 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'],
        },
    },
};
```

Пример 5.2. ACL allowing movement into a queue of only those tickets with ticket priority 5 stored in the database.

This example is very similar to the last one, but in this case only tickets in the queue "Raw" and with a priority "5 very high", both stored in the database will match. This kind of ACLs does not consider changes in the form before the ticket is really updated in the database.

```
# ticket acl
$Self->{TicketAcl}->{'100-Example-ACL'} = {
    # match properties
    PropertiesDatabase => {
        # 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'],
        },
    },
};
```

Пример 5.3. Списки прав доступа (ACL) делают невозможным закрытие заявок в очереди raw и скрывают кнопку "закрыть".

Here you can see how a ticket field (state) can be filtered with more than one possible value to select from. It is also possible to limit the actions that can be executed for a certain ticket. In this case, the ticket cannot be closed.

```
$Self->{TicketAcl}->{'101-Second-Example-ACL'} = {
    # 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 => {
            AgentTicketBounce        => 1,
            AgentTicketClose         => 0,
            AgentTicketCompose       => 1,
            AgentTicketCustomer      => 1,
            AgentTicketForward       => 1,
            AgentTicketFreeText      => 1,
            AgentTicketHistory       => 1,
            AgentTicketLink          => 1,
            AgentTicketLock          => 1,
            AgentTicketMerge         => 1,
            AgentTicketMove          => 1,
            AgentTicketNote          => 1,
            AgentTicketOwner          => 1,
            AgentTicketPending        => 1,
            AgentTicketPhone          => 1, # only used to hide the Split action
            AgentTicketPhoneInbound   => 1,
            AgentTicketPhoneOutbound  => 1,
            AgentTicketPrint          => 1,
            AgentTicketPriority       => 1,
            AgentTicketResponsible    => 1,
            AgentTicketWatcher        => 1,
            AgentTicketZoom           => 1,
            AgentLinkObject           => 1, # only used to hide the Link action
        },
    },
};
```

Пример 5.4. ACL removing always state closed successful.

This example shows how it is possible to define negative filters (the state "closed successful" will be removed). You can also see that not defining match properties for a ticket will match any ticket, i. e. the ACL will always be applied. This may be useful if you want to hide certain values by default, and only enable them in special circumstances (e. g. if the agent is in a specific group).

```
$Self->{TicketAcl}->{'102-Third-ACL-Example'} = {
    # match properties
    Properties => {
        # current ticket match properties (match always)
    },
    # return possible options
    PossibleNot => {
        # possible ticket options
        Ticket => {
            State => ['closed successful'],
        },
    },
};
```

Пример 5.5. ACL только отображает Hardware-сервисы для заявок, которые были созданы в очередях, начинающихся с "HW".

Этот пример также показывает как можно использовать регулярные выражения для сопоставления заявок и фильтрации доступных опций.

```
$Self->{TicketAcl}->{'Only-Hardware-Services-for-HW-Queues'} = {
    # match properties
    # note we don't have "Ticket => {" because there's no ticket yet
    Properties => {
    Queue => {
        Name => ['[RegExp]HW'],
        }
    },
    # return possible options
    Possible => {
        # possible ticket options
        Ticket => {
            Service => ['[RegExp]^(Hardware)'],
        },
    },
};
```

1.3. Сссылка

В нижеприведенном Сценарии представлен список всех параметров, которые могут быть использованы для списка прав доступа (ACLs).

Please see the section on ACLs in the ProcessManagement documentation for a detailed descripton of how to use ACLs for process tickets.

Пример 5.6. Ссылка отображает все возможные важные настройки ACL

```
# ticket acl
$Self->{TicketAcl}->{'200-ACL-Reference'} = {
    # match properties (current values from the form)
    Properties => {

        # the used frontend module
        Frontend => {
            Action => ['AgentTicketPhone', 'AgentTicketEmail'],
        },

        # the logged in agent
        User => {
            UserLogin => ['some login'],
            Group_rw => [
                'hotline',
            ],
            Role => [
                'admin',
            ],
            # ...
        },

        # the logged in customer
        CustomerUser => {
            UserLogin => ['some login'],
            Group_rw => [
                'hotline',
            ],
            Role => [
                'admin',
            ],
            # ...
        },

        # process properties
        Process => {
            ProcessEntityID        => ['P1'],   # the Process that the current ticket is part
 of
            ActivityEntityID       => ['A1'],   # the current Activity of the ticket
            ActivityDialogEntityID => ['AD1'],  # the current ActivityDialog that the Agent/
Customer is using
        },

        # ticket properties
        Queue => {
            Name     => ['Raw'],
            QueueID  => ['some id'],
            GroupID  => ['some id'],
            Email    => ['some email'],
            RealName => ['OTRS System'],
            # ...
        },
        Service => {
            ServiceID => ['some id'],
            Name      => ['some name'],
            ParentID  => ['some id'],
            # ...
        },
        Type => {
            ID   => ['some id'],
            Name => ['some name'],
            # ...
        },
        Priority = {
            ID   => ['some id'],
            Name => ['some name'],
            # ...
        },
        SLA = {
            SLAID    => ['some id'],
            Name     => ['some name'],
            Calendar => ['some calendar'],
            # ...
        },
        State = {
            ID       => ['some id'],
            Name     => ['some name'],
            TypeName => ['some state type name'],,
            TypeID   => ['some state type id'],
            # ...
        },
        Owner => {
            UserLogin => ['some login'],
            Group_rw => [
                'some group',
            ],
            Role => [
                'admin',
            ],
            # ...
        },
        Responsible => {
            UserLogin => ['some login'],
            Group_rw => [
                'some group',
            ],
            Role => [
                'admin',
            ],
            # ...
        },
        DynamicField => {
            # Names must be in DynamicField_<field_name> format.
            # Values in [ ... ] must always be the untranslated internal data keys
            #   specified in the dynamic field definition and
            #   not the data values shown to the user.
            DynamicField_Field1          => ['some value'],
            DynamicField_OtherField      => ['some value'],
            DynamicField_TicketFreeText2 => ['some value'],
            # ...
        },
        # alternatively, ticket properties can be specified in the ticket hash
        Ticket => {
            Queue                => ['Raw'],
            State                => ['new', 'open'],
            Priority             => ['some priority'],
            Lock                 => ['lock'],
            CustomerID           => ['some id'],
            CustomerUserID       => ['some id'],
            Owner                => ['some owner'],
            DynamicField_Field1  => ['some value'],
            DynamicField_MyField => ['some value'],
            # ...
        },
    },

    # match properties (existing values from the database)
    PropertiesDatabase => {
        # See section "Properties", the same config can be used here.
        # ...
    }

    # return possible options (white list)
    Possible => {
        # possible ticket options (white list)
        Ticket => {
            Queue => ['Hotline', 'Coordination'],
            State => ['some state'],
            Priority => ['5 very high'],
            DynamicField_Field1  => ['some value'],
            DynamicField_MyField => ['some value'],
            # ...
            NewOwner => ['some owner'],
            OldOwner => ['some owner'],
            # ...
        },

        # Limit the number of possible ActivityDialogs the Agent/Customer
        #   can use in a process ticket.
        ActivityDialog => ['AD1', 'AD3'],

        # Limit the number of possible Processes that can be started
        Process => ['P1', 'P2'],

        # possible action options (white list)
        Action => {
            AgentTicketBounce        => 1,
            AgentTicketClose         => 1,
            AgentTicketCompose       => 0,
            AgentTicketCustomer      => 0,
            AgentTicketForward       => 0,
            AgentTicketFreeText      => 1,
            AgentTicketHistory       => 1,
            AgentTicketLink          => 0,
            AgentTicketLock          => 1,
            AgentTicketMerge         => 0,
            AgentTicketMove          => 1,
            AgentTicketNote          => 1,
            AgentTicketOwner         => 1,
            AgentTicketPending       => 1,
            AgentTicketPhone         => 1, # only used to hide the Split action
            AgentTicketPhoneInbound  => 0,
            AgentTicketPhoneOutbound => 1,
            AgentTicketPrint         => 1,
            AgentTicketPriority      => 0,
            AgentTicketResponsible   => 1,
            AgentTicketWatcher       => 1,
            AgentTicketZoom          => 1,
            AgentLinkObject          => 1, # only used to hide the Link action
        },
    },
    # remove options (black list)
    PossibleNot => {
        # See section "Possible"
        # ...
    },
};
```

Примечание

While matching ACLs if CustomerUserID parameter sent, the ACL mechanism will compare the defined ACLs using the supplied CustomerUserID to gather the CustomerUser details to fill the CustomerUser hash and it also overrides the Customer information in the Ticket hash for the Properties match. On the other hand this calculations are also made for the PropertiesDatabase part, but using the Ticket Customer as the basis to gather the data.

Обратите внимание, что в Пользовательском Интерфейсе, CustomerUserID всегда отправляется с залогиненым Клиентом.

Be aware that in ticket search screens (AgentTicketSearch and CustomerTicketSearch) the only affected attributes by ACLs are the Dynamic Fields. This means that this screens you can not restrict any other attribute like ticket type, state, queue, etc.

2. Управление Процессами

2.1. Введение

This feature of OTRS allows you to model processes (work-flows) in the ticket system. The basic idea is to be able to define recurring processes, and to delegate work items to different people, as well as leading the progress of a process in different directions based on certain criteria.

2.2. Пример процесса

Давайте рассмотрим некий пример чтобы все было более наглядно и понятно. Мы будем рассматривать процесс покупки книги:

2.2.1. Запись спроса

Before an order will be placed, the demand for literature by an employee will be recorded. The following book is needed in our example:

```
Title: Prozessmanagement für Dummies
Autor: Thilo Knuppertz
ISBN: 3527703713
```

2.2.2. Approval by manager

The head of the employee's department needs to decide on the order. In case of a denial, a reason should be recorded by the manager. In case of approval, the order is passed to the purchasing department.

2.2.3. Обрабатывается отделом закупок

Purchasing now has the task to find out where the book can be ordered with the best conditions. If it is out of stock, this can be recorded in the order. In case of a successful order purchasing will record the supplier, the price and the delivery date.

2.2.4. Processing by the mail room

The shipment will arrive at the company. The incoming goods department checks the shipment and records the date of receipt. Now the employee will be informed that their order has arrived and is ready to be collected.

2.3. Выполнение примера

If we assume that a ticket acts in this work-flow like an accompanying document that can receive change notes, we already have a clear picture of process tickets.

From the analysis of the example process we can identify the following necessary items:

- Possibilities to record data, let's call them *Activity Dialogs*,
- Checks which can react to changed data automatically, let's call them *Transitions*,
- изменения, которые могут быть применены для обработки заявки после успешного перехода обработки заявки, назовем их *Transition Actions*.

Также нам необходим дополнительный элемент, который не так очевиден:

- A possibility to offer more than just one Activity Dialog to be available. In our example this is needed when the manager must have the choice between "Approve" and "Deny". Let's call this *Activity*.

Now, with Activities, Activity Dialogs, Transitions and Transition Actions we have the necessary tools to model the individual steps of our example. What's still missing is an area where for each work-flow the order of the steps can be specified. Let's call this *Process*. To be able to refer to all these entities later, we will assign to them an abbreviation in parentheses. This abbreviation is based on an internal identification mechanism called EntityIDs.

EntityIDs согласовываются с одной или двумя буквами (в зависимости от части процесса), а затем порядковый номер, например:

- Процесс: 'P1', 'P2' ... 'Pn'.
- Деятельность: 'A1', 'A2' ... 'An'.
- Диалог Активности: 'AD1', 'AD2' ... 'ADn'.
- Переход: 'T1', 'T2' ... 'Tn'.
- Переход действий: 'TA1', 'TA2' ... 'TAn'.

Перед созданием процесса и его частей необходимо подготовить систему. Нужно определить некоторые Очереди, Клиентов и Динамические Поля, а также установить некоторые опции в файле SysConfig.

Создать следующие Очереди:

- Management
- Employees
- Покупка
- Почтовое отделение

Создает следующих Пользователей:

- Manager
- Сотрудник

Создать следующие Динамически Поля:

- Название

Метка	Название
Type	Text
Объект	Заявк

- Автор

Метка	Автор
Type	Text
Объект	Заявк

- ISBN

Метка	ISBN
Type	Text
Объект	Заявк

- Состояние

Метка	Состояние
Type	Dropdown
Объект	Заявк
Возможные значения	• Утверждение • Отказано в утверждении • Утвержденный • Заказ отменен • Заказ размещен • Посылка получена

Note: Please use this exactly this possible values for "Key" and "Value" in the Dynamic Field setup.

- Поставщик

Метка	Поставщик
Type	Text
Объект	Заявк

- Цена

Метка	Цена
Type	Text
Объект	Заявк

- DeliveryDate

Метка	Дата доставки
Type	Date
Объект	Заявк

- DateOfReceipt

Метка	Date Of Receipt
Type	Date
Объект	Заявк

Устанавливает следующие параметры SysConfig:

- 'Ticket::Responsible': Yes
- 'Ticket::Frontend::AgentTicketZoom###ProcessWidgetDynamicFieldGroups':

Ключ:	***Содержание:***
Книга	Название, Автор, ISBN
Общий	Состояние
Заказ	Цена, Поставщик, Дада доставки
Отгрузка	DateOfReceipt

- 'Ticket::Frontend::AgentTicketZoom###ProcessWidgetDynamicField':

Ключ:	***Содержание:***
Автор	1
DateOfReceipt	1
DeliveryDate	1
ISBN	1
Цена	1
Состояние	1
Поставщик	1
Название	1

Now lets start with the real Process Management stuff. In the next step, we will define the individual entities that we need.

2.3.1. Процесс (в качестве контейнера)

To create a new process is necessary to click on the "Process Management" link in the System Administration box in the Admin panel, this will lead to the Process Management Overview screen. After the creation of the process we can create all other entities (or process parts).

Примечание

Деятельности, Диалоги Активности, Переходы и Действия переходов определенные в одном процессе будут доступны для всех процессов системы.

Рисунок: Страница Администратора OTRS - Администрирование системы.

Нажмите на действие "Создать Новый Процесс" в окне Действий.

Figure: Создать кнопку Новый Процесс.

Заполняем информацию о процессе, указываем Название Процесса и его Описание, а статус оставим со значением "inactive", пока не закончим все задачи. Сохраняем процесс.

Рисунок Добавить новый процесс.

2.3.2. Диалоги Действий

Click on the new process name in the Process Management Overview Screen, then in the "Available Process Elements" click in "Activity Dialogs" (this action will expand the activity dialog options and will collapse all others doing and accordion like effect), then click on "Create New Activity Dialog".

Рисунок: Создать кнопку Новый Диалог Деятельности.

In the opened popup screen fill the "Activity dialog Name" as well as the "Description (short)" fields, for this example we will leave all other fields as the default, to assign fields to the Activity Dialog simple drag the required field from the "Available Fields" pool and drop into the "Assigned Fields" pool. The order in the "Assigned Fields" pool is the order as the fields will have in the screen, to modify the order simply drag and drop the filed within the pool to rearrange it in the correct place.

Figure: Добавить новый Диалог Деятельности.

As soon as the fields are dropped into the "Assigned Fields" pool another popup screen is shown with some details about the field, we will leave the default options and only for Article fields we should make sure that the ArticleType field is set to "note-internal".

Figure: Edit field details (Article).

Если все поля привязаны, нажмите на кнопку отправить в главном всплывающем окне чтобы изменения вступили в силу.

In this example we will use Article field for comments, but another option could be to create a TextArea type Dynamic Field, the rest of the mentioned fields in the lines below are the Dynamic Fields that we define before.

Обратите вниманиче что на этой странице все Динамические Поля имеют префикс "DynamicField_" как в "DynamicField_Title", не путайте с полем "Название", которое является Названием Заявки.

Создать следующие Диалоги Активности:

- "Запись спроса" (AD1)

  An Activity Dialog that contains all the required fields for the data to be collected for the order (Title, Author and ISBN), and a Status field with the possibility to choose "Approval".

- "Отказано в утверждении" (AD2)

  An Activity Dialog with a comment field (Article) and a Status field with the option "Approval denied".

- "Утвержден" (AD3)

  Here we just need the Status field with the option "Approved".

- "Order denied" (AD4)

  An activity dialog which makes it possible for purchasing to reject an impossible order (book out of stock). Here we also need a comment field and the Status field with the option "Order denied".

- "Заказ размещен" (AD5)

  An activity dialog with the fields Supplier, Price and Delivery date for purchasing and the Status field with the option "Order placed".

- "Поставка получена" (AD6)

  An activity for the mail room with a field for the Date of receipt and the Status field with the option "Shipment received".

Чтобы ограничить поле состояние для каждого из диалогов активности нам необходимо добавить некоторые ACL-права в файл Kernel/Config.pm или в новый perl-файл в Kernel/Config/Files.

```
$Self->{TicketAcl}->{'P1-AD1-1'} = {
    Properties => {
        Process => {
            ActivityDialogEntityID => ['AD1'],
        },
    },
    Possible => {
        Ticket => {
            DynamicField_Status => ['Approval'],
        },
    },
};

$Self->{TicketAcl}->{'P1-AD2-1'} = {
    Properties => {
        Process => {
            ActivityDialogEntityID => ['AD2'],
        },
    },
    Possible => {
        Ticket => {
            DynamicField_Status => ['Approval denied'],
        },
    },
};

$Self->{TicketAcl}->{'P1-AD3-1'} = {
    Properties => {
        Process => {
            ActivityDialogEntityID => ['AD3'],
        },
    },
    Possible => {
        Ticket => {
            DynamicField_Status => ['Approved'],
        },
    },
};

$Self->{TicketAcl}->{'P1-AD4-1'} = {
    Properties => {
        Process => {
            ActivityDialogEntityID => ['AD4'],
        },
    },
    Possible => {
        Ticket => {
            DynamicField_Status => ['Order denied'],
        },
    },
};

$Self->{TicketAcl}->{'P1-AD5-1'} = {
    Properties => {
        Process => {
            ActivityDialogEntityID => ['AD5'],
        },
    },
    Possible => {
        Ticket => {
            DynamicField_Status => ['Order placed'],
        },
    },
};

$Self->{TicketAcl}->{'P1-AD6-1'} = {
    Properties => {
        Process => {
            ActivityDialogEntityID => ['AD6'],
        },
    },
    Possible => {
        Ticket => {
            DynamicField_Status => ['Shipment received'],
        },
    },
};
```

2.3.3. Переходы

В "Доступные Элементы Процесса" щелкните "Переходы", а затем кликните на "Создать новый переход".

Figure: Создать кнопку Новый Переход.

In the opened popup screen fill the "Transition Name", then in the conditions, for this examples we will use just one condition and just one field, for both we can leave the Type of Linking as "and" and we will use the filed match type value as "String".

Рисунок: Добавить новый ПереходAdd new Transition.

Когда все условия установлены, нажмите кнопку Отправить, чтобы сохнанить изменения.

Создать следующие Переходы:

- "Утверждение" (T1)

  A transition which checks if the Status field is set to "Approval".

- "Отказано в утверждении" (T2)

  A transition which checks if the Status field is set to "Approval denied".

- "Утвержден" (T3)

  A transition which checks if the Status field is set to "Approved".

- "Заказ отменен" (T4)

  A transition which checks if the Status field is set to "Order denied".

- "Заказ размещен" (T5)

  A transition which checks if the Status field is set to "Order placed".

- "Поставка получена" (T6)

  A transition which checks if the Status field is set to "Shipment received".

2.3.4. Transition Actions

Click on "Transition Actions" in the "Available Process Elements", then click on "Create New Transition Action".

Figure: Create New Transition Action button.

В открывшемся всплывающем окне заполните поле "Название Перехода" и "Модуль Деятельности Перехода", а затем добавьте требуемые и опциональные параметры, названия и значения.

Все Модули Переходов Действий находятся в Kernel/System/ProcessManagement/TransitionAction и ниже приводится список Переходов Действий включенных в этот релиз

- DynamicFieldSet
- TicketArticleCreate
- TicketCustomerSet
- TicketLockSet
- TicketOwnerSet
- TicketQueueSet
- TicketResponsibleSet

- TicketServiceSet
- TicketSLASet
- TicketStateSet
- TicketTitleSet
- TicketTypeSet

Each module has its own and different parameters. Please review the module documentation to learn all require and optional parameters.

Рисунок: Добавить новый переход деятельности.

Как только все параметры и значения установлены, нажмите кнопку отправить чтобы изменения вступили в силу.

Create the following Transitions Actions:

- "Move the process ticket into the 'Management' queue" (TA1)

  This action is supposed to be executed when the Transition "Approval" (T1) applied.

- "Change ticket responsible to 'manager'" (TA2)

  Будет выполнено когд применен Переход "Approval" (T1).

- "Move the process ticket into the 'Employees' queue"" (TA3)

  Будет выполнено когда:

  - Применение Перехода "Отказано в утверждении" (T2)
  - Применение Перехода "Order denied" (T4)
  - Применение Перехода "Поставка получена" (T6)

- "Change ticket responsible to 'Employee'" (TA4)

  Будет выполнено когда:

  - Применен переход "Отказано в утверждении" (T2)
  - Применен переход "Поставка получена" (T6)
  - Применен переход The transition applied

- "Move process ticket into the 'Purchasing' queue" (TA5)

  To be executed when the transition "Approved" (T3) applied.

- "Move process ticket into the 'Post office' queue" (TA6)

  To be executed when the transition "Order placed" (T5) applied.

- "Close ticket successfully" (TA7)

  Будет выполнено когда:

  - Применен переход The transition applied

- "Close ticket unsuccessfully" (TA8)

  Будет выполнено когда:

  - Применен переход "Отказано в утверждении" (T2)
  - Применен переход "Поставка получена" (T6)

As you can see, there are places where the same Transition Actions should be executed. Therefore it is reasonable to make it possible to link Transition Actions freely with Transitions to be able to reuse them.

2.3.5. Деятельность

We chose the approach to see Activities as a basket which can contain one or more Activity Dialogs.

Кликните на "Деятельности" в меню "Доступные Элементы Процесса", а затем кликните на "Создать новую Деятельность".

Figure: Create New Activity button.

In the opened popup screen fill the "Activity Name", Then drap the required Activity Dialogs from the "Available Activity Dialogs" pool, and drop then into to the "Assigned Activity Dialogs" pool. This dialogs will be presented (in the ticket zoom screen) in the same order as it is defined on this screen translating from top to bottom, to left to right.

Такой порядок особенно важен в первой Деятельности, так как первый Диалог Деятельности для этой Деятельности является единственным, который находится при запуске процесса

Создать следующие Деятельности:

- "Recording the demand" (A1)

  Contains the Activity Dialog "Recording the demand" (AD1)

- "Approval" (A2)

  Содержит Диалоги Активности "Отказано в одобрении" (AD2) а также "Утвержден" (AD3)

- "Order" (A3)

  Содержит Диалоги Активности "Заказ отклонен" (AD4) а также "Заказ размещен" (AD5)

- "Incoming" (A4)

  Contains the Activity Dialog "Shipment received" (AD6)

- "Process complete" (A5): This is an Activity without possible Activity Dialogs. It will be set after "Approval denied", "Order denied" or "Shipment received" and represents the end of the process.

Теперь мы можем видеть, что Деятельность точно определяет сосотояние процесса Заявки. После успешного Перехода Заявка перемещается из одного вида деятельности в другой.

2.3.6. Book ordering process Path

Let us conclude our example with the last missing piece in the puzzle, the Process as the a flow describer. In our case this is the whole ordering work-flow. Other processes could be office supply ordering or completely different processes.

The process has a starting point which consists of the start Activity and the start Activity Dialog. For any new book order, the start Activity Dialog (first Activity Dialog for the first Activity) is the first screen that is displayed. If this is completed and saved, the Process ticket will be created and can follow the configured work-flow.

The process also contains the directions for how the process ticket can move through the Process. Let's call this the "Path". It consists of the start Activity, one or more Transitions (possibly with Transition Actions), and other Activities.

Assuming that the Activities has already assigned their Activity Dialogs drag an Activity from the accordion (in the left part of the screen) and drop it into the canvas area (below process information). Notice that an arrow from the process start (green circle) to the Activity is placed automatically. (This is the first Activity and its first Activity Dialog is the first screen that will be shown when the process starts).

Figure: Drag first Activity into the canvas.

Next, drag another Activity into the canvas too. now we will have two Activities in the canvas the first one is connected to the start point and the second has no connections, you can hover the mouse over each activity to reveal their own Activity Dialogs.

Figure: Drag second Activity into the canvas.

Then let's create the "Path" (connection) between this two Activities, for this we will use the Transitions, Click on Transitions in the accordion drag a Transition and drop it inside the first Activity, notice that the Activity change its color indicating that the Transition is attached, as soon as the Transition is dropped the end point of the Transition arrow will be placed next to the process start point. Drag the Transition arrow end point and drop it inside the other Activity to create the connection between the Activities.

Figure: Drag a Transition into the canvas.

Now that the "Path" between the Actions is defined, then we need to assign the Transition Actions to the Transition, double click the Transition label (in the canvas), this will open a new popup window.

Figure: Connect Activities using Transitions.

Drag the needed Transition Actions from Available Transition Actions pool and drop them into the Assigned Transition Actions pool and click on submit button.

Figure: Assign Transition Actions.

Then back in the main process edit screen click on save button below the canvas to save all other changes.

Complete the "path"" adding the following Activities, Transitions and Transition Actions:

Запись спроса до значения "Approval"

- Starting point: Activity: Recording the demand (A1)
- Possible Transition: Approval (T1)
  - If the condition of this activity is fulfilled, the ticket will move to Activity: Approval (A2)
  - Кроме того выполнится следующее действие TransitionActions:
    - "Moving the ticket into the 'Management' Queue" (TA1)
    - "Set ticket responsible to 'manager'" (TA2)

The Activity: "Recording the demand" (A1) is a defined step of the process ticket, where there is the possibility for the Transition: "Approval" (T1). If this applies, the ticket will move to the next Activity: "Approval" (A2), and the Transition Actions: "Moving the ticket into the 'Management' Queue" (TA1) and "Set ticket responsible to 'manager'" (TA2) are executed. In the Activity: "Approval" (A2), the Activity Dialogs: "Approval denied" (AD2) and "Approved" (AD3) are available.

Утверждение

- Starting Point: Activity "Approval" (A2)
- Возможные Транзакции:
  - "Отказано в утверждении" (T2)
    - If this matches, the process ticket will move to Activity:"Process complete" (A5).
    - Additionally, the following Transition Actions are executed:
      - "Move process ticket to the 'Employees' Queue" (TA3)
      - "Set ticket responsible to 'Employee'" (TA4)
      - "Close ticket unsuccessfully" (TA8)
  - "Утвержден" (T3)
    - If this matches, the process ticket will move to Activity: "Order" (A3).
    - Additionally, the following Transition Action is executed:
      - "Move process ticket to 'Purchasing' Queue" (TA5)

We can see that from the current Activity, which defines a step of the process ticket, there are one or more possibilities for Transition which have exactly one target Activity (and possibly one or more Transition Actions).

Заказ

- Starting Point: Activity "Order" (A3)
- Возможные Транзакции:
  - "Заказ отменен" (T4)
    - If this matches, the process ticket will move to Activity:"Process complete" (A5).
    - Additionally, the following Transition Actions are executed:
      - "Move process ticket to the 'Employees' Queue" (TA3)
      - "Set ticket responsible to 'Employee'" (TA4)
      - "Close ticket unsuccessfully" (TA8)
  - "Заказ размещен" (T5)
    - If this matches, the process ticket will move to Activity: "Incoming" (A5).
    - Additionally, the following Transition Action is executed:
      - "Move process ticket to 'Post office' Queue" (TA6)

Incoming

- Starting Point: Activity "Incoming" (A4)
- Возможные Транзакции:
  - "Поставка получена" (T6)
    - If this matches, the process ticket will move to Activity:"Process complete" (A5).
    - Additionally, the following Transition Actions are executed:
      - "Move process ticket to the 'Employees' Queue" (TA3)
      - "Set ticket responsible to 'Employee'" (TA4)
      - "Close ticket successfully" (TA7)

Полный путь к Процессу покупкпи книги будет иметь следующий вид:

Figure: Book ordering complete process path.

After you finish the process path please click on "Save" button in the lower part of the canvas and then click on "Synchronize All Processes" button. This will gather all processes information form the Database and create a cache file (in Perl language). This cache file is acctually the processes configuration that the system will use to create or use process tickets.

Any change that is made of the process (in the GUI) will require to re-synchronize the cache file in order to get the change reflected in the system.

It is also possible to import the whole process from a YAML file, but it is still necessary to create all Dynamic Fields, Users, Queues, etc that are needed by each process before the import.

Notice that if the process requires the use of ACLs those are also needed to be set manually.

The follwing is the complete YAML file for the book ordering process example:

```
---
Activities:
  A1:
    ActivityDialogs:
      - AD1
    ChangeTime: 2012-11-23 14:49:22
    Config:
      ActivityDialog:
        1: AD1
    CreateTime: 2012-11-23 11:49:38
    EntityID: A1
    ID: 151
    Name: Recording the demand
  A2:
    ActivityDialogs:
      - AD2
      - AD3
    ChangeTime: 2012-12-13 00:55:12
    Config:
      ActivityDialog:
        1: AD2
        2: AD3
    CreateTime: 2012-11-23 11:50:11
    EntityID: A2
    ID: 152
    Name: Approval
  A3:
    ActivityDialogs:
      - AD4
      - AD5
    ChangeTime: 2012-11-23 18:12:14
    Config:
      ActivityDialog:
        1: AD4
        2: AD5
    CreateTime: 2012-11-23 11:50:35
    EntityID: A3
    ID: 153
    Name: Order
  A4:
    ActivityDialogs:
      - AD6
    ChangeTime: 2012-11-23 18:12:35
    Config:
      ActivityDialog:
        1: AD6
    CreateTime: 2012-11-23 11:51:00
    EntityID: A4
    ID: 154
    Name: Incoming
  A5:
    ActivityDialogs: []
    ChangeTime: 2012-11-23 11:51:33
    Config: {}
    CreateTime: 2012-11-23 11:51:33
    EntityID: A5
    ID: 155
    Name: Process complete
ActivityDialogs:
  AD1:
    ChangeTime: 2012-12-06 02:16:21
    Config:
      DescriptionLong: ''
      DescriptionShort: Recoding the demand
      FieldOrder:
        - DynamicField_Author
        - DynamicField_ISBN
        - DynamicField_Title
        - DynamicField_Status
      Fields:
        DynamicField_Author:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
        DynamicField_ISBN:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
        DynamicField_Status:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
        DynamicField_Title:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
      Interface:
        - AgentInterface
      Permission: ''
      RequiredLock: 0
      SubmitAdviceText: ''
      SubmitButtonText: ''
    CreateTime: 2012-11-23 14:34:43
    EntityID: AD1
    ID: 154
    Name: Recording the demand
  AD2:
    ChangeTime: 2012-11-23 14:57:41
    Config:
      DescriptionLong: ''
      DescriptionShort: Approval denied
      FieldOrder:
        - Article
        - DynamicField_Status
      Fields:
        Article:
          Config:
            ArticleType: note-internal
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
        DynamicField_Status:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
      Interface:
        - AgentInterface
      Permission: ''
      RequiredLock: 0
      SubmitAdviceText: ''
      SubmitButtonText: Deny Request
    CreateTime: 2012-11-23 14:36:39
    EntityID: AD2
    ID: 155
    Name: Approval denied
```

```
AD3:
  ChangeTime: 2012-12-14 03:14:23
  Config:
    DescriptionLong: ''
    DescriptionShort: Approved
    FieldOrder:
      - DynamicField_Status
    Fields:
      DynamicField_Status:
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
    Interface:
      - AgentInterface
    Permission: ''
    RequiredLock: 0
    SubmitAdviceText: ''
    SubmitButtonText: Approve Request
  CreateTime: 2012-11-23 14:37:35
  EntityID: AD3
  ID: 156
  Name: Approved
AD4:
  ChangeTime: 2012-11-23 14:58:52
  Config:
    DescriptionLong: ''
    DescriptionShort: Order rejected
    FieldOrder:
      - Article
      - DynamicField_Status
    Fields:
      Article:
        Config:
          ArticleType: note-internal
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
      DynamicField_Status:
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
    Interface:
      - AgentInterface
    Permission: ''
    RequiredLock: 0
    SubmitAdviceText: ''
    SubmitButtonText: Reject Order
  CreateTime: 2012-11-23 14:38:48
  EntityID: AD4
  ID: 157
  Name: Order rejected
AD5:
  ChangeTime: 2012-12-06 02:20:12
  Config:
    DescriptionLong: ''
    DescriptionShort: Order placed
    FieldOrder:
      - DynamicField_DeliveryDate
      - DynamicField_Price
      - DynamicField_Supplier
      - DynamicField_Status
    Fields:
      DynamicField_DeliveryDate:
        DefaultValue: ''
```

```
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
        DynamicField_Price:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
        DynamicField_Status:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
        DynamicField_Supplier:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
      Interface:
        - AgentInterface
      Permission: ''
      RequiredLock: 0
      SubmitAdviceText: ''
      SubmitButtonText: Place Order
    CreateTime: 2012-11-23 14:41:28
    EntityID: AD5
    ID: 158
    Name: Order placed
  AD6:
    ChangeTime: 2012-11-23 14:42:43
    Config:
      DescriptionLong: ''
      DescriptionShort: Shipment received
      FieldOrder:
        - DynamicField_DateOfReceipt
        - DynamicField_Status
      Fields:
        DynamicField_DateOfReceipt:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
        DynamicField_Status:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
      Interface:
        - AgentInterface
      Permission: ''
      RequiredLock: 0
      SubmitAdviceText: ''
      SubmitButtonText: ''
    CreateTime: 2012-11-23 14:42:43
    EntityID: AD6
    ID: 159
    Name: Shipment received
Process:
  Activities:
    - A1
    - A2
    - A3
    - A4
    - A5
  ChangeTime: 2012-12-06 02:31:59
  Config:
    Description: The process to order a book
    Path:
      A1:
        T1:
          ActivityEntityID: A2
          TransitionAction:
            - TA2
            - TA1
      A2:
        T2:
          ActivityEntityID: A5
          TransitionAction:
            - TA3
            - TA4
            - TA8
        T3:
          ActivityEntityID: A3
          TransitionAction:
            - TA5
      A3:
        T4:
          ActivityEntityID: A5
          TransitionAction:
            - TA3
            - TA4
            - TA8
        T5:
          ActivityEntityID: A4
          TransitionAction:
            - TA6
      A4:
        T6:
          ActivityEntityID: A5
          TransitionAction:
            - TA3
            - TA4
            - TA7
      A5: {}
    StartActivity: A1
    StartActivityDialog: AD1
  CreateTime: 2012-11-23 11:45:12
  EntityID: P1
  ID: 94
  Layout:
    A1:
      left: 172
      top: 63
    A2:
      left: 402
      top: 156
    A3:
      left: 649
      top: 255
    A4:
      left: 774
      top: 391
    A5:
      left: 194
      top: 410
  Name: Book ordering
  State: Active
  StateEntityID: S1
  TransitionActions:
    - TA1
    - TA2
    - TA3
    - TA4
    - TA8
```

```
    - TA5
    - TA3
    - TA4
    - TA8
    - TA6
    - TA3
    - TA4
    - TA7
  Transitions:
    - T1
    - T2
    - T3
    - T4
    - T5
    - T6
TransitionActions:
  TA1:
    ChangeTime: 2012-11-23 16:01:37
    Config:
      Config:
        Queue: Management
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet
    CreateTime: 2012-11-23 15:50:59
    EntityID: TA1
    ID: 61
    Name: Move the process ticket into the "Management" queue
  TA2:
    ChangeTime: 2012-11-23 16:02:12
    Config:
      Config:
        Responsible: manager
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketResponsibleSet
    CreateTime: 2012-11-23 15:58:22
    EntityID: TA2
    ID: 62
    Name: Change ticket responsible to "manager"
  TA3:
    ChangeTime: 2012-11-24 14:27:02
    Config:
      Config:
        Queue: Employees
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet
    CreateTime: 2012-11-23 16:02:54
    EntityID: TA3
    ID: 63
    Name: Move the process ticket into the "Employees" queue
  TA4:
    ChangeTime: 2012-11-23 16:04:06
    Config:
      Config:
        Responsible: Employee
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketResponsibleSet
    CreateTime: 2012-11-23 16:04:06
    EntityID: TA4
    ID: 64
    Name: Change ticket responsible to "Employee"
  TA5:
    ChangeTime: 2012-12-06 02:18:34
    Config:
      Config:
        Queue: Purchasing
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet
    CreateTime: 2012-11-23 16:04:54
    EntityID: TA5
    ID: 65
    Name: Move process ticket into the "Purchasing" queue
  TA6:
    ChangeTime: 2012-12-06 02:18:48
    Config:
      Config:
        Queue: Post office
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet
    CreateTime: 2012-11-23 16:06:20
    EntityID: TA6
    ID: 66
    Name: Move process ticket into the "Post office" queue
  TA7:
    ChangeTime: 2012-12-06 02:29:55
    Config:
      Config:
        State: closed successful
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketStateSet
    CreateTime: 2012-12-06 02:29:27
    EntityID: TA7
    ID: 67
    Name: Close ticket successfully
  TA8:
    ChangeTime: 2012-12-06 02:31:12
    Config:
      Config:
        State: closed unsuccessful
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketStateSet
    CreateTime: 2012-12-06 02:31:12
    EntityID: TA8
    ID: 68
    Name: Close ticket unsuccessfully
Transitions:
  T1:
    ChangeTime: 2012-11-23 15:12:20
    Config:
      Condition:
        1:
          Fields:
            DynamicField_Status:
              Match: Approval
              Type: String
          Type: and
      ConditionLinking: and
    CreateTime: 2012-11-23 11:53:52
    EntityID: T1
    ID: 94
    Name: Approval
  T2:
    ChangeTime: 2012-11-23 15:12:50
    Config:
      Condition:
        1:
          Fields:
            DynamicField_Status:
              Match: Approval denied
              Type: String
          Type: and
      ConditionLinking: and
    CreateTime: 2012-11-23 11:54:26
    EntityID: T2
    ID: 95
    Name: Approval denied
  T3:
    ChangeTime: 2012-11-23 15:13:29
    Config:
      Condition:
        1:
          Fields:
            DynamicField_Status:
              Match: Approved
              Type: String
          Type: and
      ConditionLinking: and
    CreateTime: 2012-11-23 11:54:54
    EntityID: T3
    ID: 96
    Name: Approved
  T4:
    ChangeTime: 2012-11-23 15:14:08
    Config:
      Condition:
        1:
          Fields:
            DynamicField_Status:
              Match: Order denied
              Type: String
          Type: and
      ConditionLinking: and
    CreateTime: 2012-11-23 11:55:25
    EntityID: T4
    ID: 97
    Name: Order denied
  T5:
    ChangeTime: 2012-11-23 18:30:33
    Config:
      Condition:
        1:
          Fields:
            DynamicField_Status:
              Match: Order placed
              Type: String
          Type: and
      ConditionLinking: and
    CreateTime: 2012-11-23 11:56:15
    EntityID: T5
    ID: 98
    Name: Order placed
  T6:
    ChangeTime: 2012-11-23 15:15:30
    Config:
      Condition:
        1:
          Fields:
            DynamicField_Status:
              Match: Shipment received
              Type: String
          Type: and
      ConditionLinking: and
    CreateTime: 2012-11-23 11:56:48
    EntityID: T6
    ID: 99
    Name: Shipment received
```

2.4. Process configuration reference

2.4.1. Process

A Process models the path of a workflow/process. The waypoints on this path can be Activities or Transitions, we'll talk about these later.

2.4.1.1. Process configuration

The Process configuration can be done in the file Kernel/Config.pm but it is strongly recommended to create new files like Kernel/Config/Files/MyProcess.pm. notice that the

GUI generates the file Kernel/Config/File/ZZZProcessManagement please avoid to use that filename, otherwise it will be overwritten when you sync processes. Let's see an example process configuration (from process cache file):

```
$Self->{'Process'} = {
    'P1' => {
        Name                => 'Book order',
        CreateTime          => '16-02-2012 13:37:00',
        CreateBy            => '1',
        ChangeTime          => '17-02-2012 13:37:00',
        ChangeBy            => '1',
        State               => 'Active',
        StartActivity       => 'A1',
        StartActivityDialog => 'AD1',
        Path => {
            'A1' => {
                'T1' => {
                    ActivityEntityID => 'A2',
                },
            },
            'A2' => {
                'T2' =>  {
                    ActivityEntityID => 'A3',
                },
            },
        },
    },
    'P2' => {
        Name                => 'IT order',
        CreateTime          => '26-02-2012 13:37:00',
        CreateBy            => '1',
        ChangeTime          => '27-02-2012 13:37:00',
        ChangeBy            => '1',
        State               => 'Active',
        StartActivity       => 'A2',
        StartActivityDialog => 'AD2',
        Path => {
            'A2' => {
                'T3' => {
                    ActivityEntityID => 'A4',
                },
            },
        },
    }
};
```

2.4.1.2. Название

The name of the process, this can be selected by the agent when creating a new process ticket.

2.4.1.3. CreateTime

The time when the process was created.

2.4.1.4. CreateBy

The UID of the user creating the process.

2.4.1.5. ChangeTime

The time when the process was changed.

2.4.1.6. ChangeBy

The UID of the user who made the last change to the process.

2.4.1.7. State

Defines the state of a process. Possible values:

- 'Active' are all processes which can be used in new process tickets.
- 'FadeAway' are processes which cannot be selected any more for new tickets, but existing tickets still can use the process.
- 'Inactive' processes are deactivated and cannot be used for new or existing tickets.

2.4.1.8. StartActivity

When creating a new process ticket, a StartActivity must be defined. As soon as the ticket is created, this Activity will be set and used as the base for the first transition checks.

2.4.1.9. StartActivityDialog

For new process tickets, a StartActivityDialog must be defined. This will be shown when creating a new process ticket (after the process was selected). At this point, the ticket does not exist yet, it will be created after submitting the StartActivityDialog.

2.4.1.10. Path

The Path contains the structure of the Activities, and the possible Transitions between them, for the current process. And also the Transition Actions that happens when transitioning . This controls the way that a process ticket can take. Example:

```
'A1' => {
    'T1' => {
        ActivityEntityID => 'A2',
    },
    'T2' => {
        ActivityEntityID => 'A3',
    },
    'T3' => {
        ActivityEntityID => 'A4',
        TransitionAction => ['TA1', 'TA2'],
    },
},
```

If a process ticket is in Activity 'A1', it has three possible ways to get to another Activity. In the Transitions 'T1' to 'T3', conditions are defined, that a process ticket must fulfill to move (transit) to another Activity.

If in this case all the values of the process ticket and its dynamic fields that are needed for the Transition 'T2' are correct, the ticket will be moved from Activity 'A1' to 'A3'. After an ActivityDialog is submitted, or any other change is made to a ticket, it will be checked for possible Transitions from the current Activity. If multiple Transitions are possible, the first one will be used (based on nummerical sorting of the TransitionIDs).

Additionally, it is possible to assign Transition Actions to Transitions in the Path configuration. These are modules which are executed after a successful Transition. They have to be specified in array form as in the example, we'll talk about the details later.

2.4.2. Activity

An Activity contains one or more Activity Dialogs and models a 'step' in the process. All Activity Dialogs of the current Activity are displayed in the ticket zoom and can be used until the conditions of a Transition are fulfilled.

2.4.2.1. Activity configuration

Let's see an example activity configuration:

```
$Self->{'Process::Activity'} =
{
    'A1' => {
        Name       => 'Activity 1 optional',
        CreateTime => '16-02-2012 13:37:00',
        CreateBy   => '1',
        ChangeTime => '17-02-2012 13:37:00',
        ChangeBy   => '1',
        ActivityDialog => {
            1 => 'AD1',
        },
    },
    'A2' => {
        Name       => 'Activity 2 optional',
        CreateTime => '16-02-2012 13:37:00',
        CreateBy   => '1',
        ChangeTime => '17-02-2012 13:37:00',
        ChangeBy   => '1',
        ActivityDialog => {
            1 => 'AD5',
            2 => 'AD6',
            3 => 'AD1',
        },
    },
};
```

2.4.2.2. Название

The name of the activity.

2.4.2.3. CreateTime

The time when it was created.

2.4.2.4. CreateBy

UID of the user who created the Activity.

2.4.2.5. ChangeTime

The last time when it was changed.

2.4.2.6. ChangeBy

UID of the last user who changed the Activity.

2.4.2.7. ActivityDialog

Activity Dialog contains the list of Activity Dialogs which are available in this Activity. All Activity Dialogs of the current Activity are displayed in the ticket zoom. Their order is set by the order in the configuration, here 'AD5' is shown before 'AD6' and 'AD1'.

2.4.3. ActivityDialog

An Activity Dialog is a particular screen and can be used in different Activities.

2.4.3.1. ActivityDialog configuration

Let's see an example config

```
$Self->{'Process::ActivityDialog'} = {
    'AD1' => {
        Name             => 'ActivityDialog 1 optional',
        DescriptionShort => 'Basic info',
        DescriptionLong  => 'Please insert the necessesary basic information for IT orders',
        CreateTime       => '28-02-2012 13:37:00',
        CreateBy         => '1',
        ChangeTime       => '29-02-2012 13:37:00',
        ChangeBy         => '1',
        Fields => {
            PriorityID => {
                DescriptionShort => 'Priority ID',
                DescriptionLong  => 'Enter the priority here',
                Display          => 2,
            },
        },
        FieldOrder       => [ 'PriorityID' ],
        SubmitAdviceText => 'Note: If you submit the form...',
        SubmitButtonText => 'Send request',
    },
    'AD2' => {
        Name => 'ActivityDialog 2 optional',
        DescriptionShort => 'Basic info',
        DescriptionLong  => 'Please insert the necessesary basic information for Book orders',
        CreateTime       => '28-02-2012 13:37:00',
        CreateBy         => '1',
        ChangeTime       => '29-02-2012 13:37:00',
        ChangeBy         => '1',
        Fields => {
            StateID => {
                DescriptionShort => 'State ID',
                DescriptionLong  => 'Enter the state here',
                Display          => 2,
                DefaultValue     => '2',
            },
            Queue => {
                DescriptionShort => 'Queue ID',
                DescriptionLong  => 'Enter the queue here',
                Display          => 2,
                DefaultValue     => 'Raw',
            },
            Title => {
                DescriptionShort => 'Title',
                DescriptionLong  => 'Enter the title here',
                Display          => 1,
                DefaultValue     => 'Default Title',
            },
            DynamicField_Anzahl => {
                DescriptionShort    => 'Amount',
                DescriptionLong     => 'Enter the amount here',
                Display             => 2,
                DefaultValue        => '4',
            },
        },
        FieldOrder       => [ 'DynamicField_Anzahl', 'StateID', 'Queue', 'Title' ],
        SubmitAdviceText => 'Note: If you submit the form...',
        SubmitButtonText => 'Send request',
    },
};
```

2.4.3.2. Название

Name of the Activity Dialog.

2.4.3.3. CreateTime

Time when it was created.

2.4.3.4. CreateBy

UID of the user who created this Activity Dialog.

2.4.3.5. ChangeTime

Last time when it was changed.

2.4.3.6. ChangeBy

UID of the last user who changed this Activity Dialog.

2.4.3.7. Fields

Contains all fields which can be displayed in this Activity Dialog. The following fields can currently be used:

```
    Title
    State
    StateID
    Priority
    PriorityID
    Lock
    LockID
    Queue
    QueueID
    Customer
    CustomerID
    CustomerNo
    CustomerUserID
    Owner
    OwnerID
    Type
    TypeID
    SLA
    SLAID
    Service
    ServiceID
    Responsible
    ResponsibleID
    PendingTime
    DynamicField_$FieldName  # for all dynamic fields
```

Example of a single field configuration:

```
StateID => {
    DescriptionShort => 'State ID',
    DescriptionLong  => 'Enter the state here',
    Display          => 2,
    DefaultValue     => '2',
},
```

The field "Article" is a special case. If it is present in a "Fields" configuration, the Activity Dialog will contain a complete Richtext editor with subject field and attachment handling. The entered text will then be added to the ticket as an article and sent by email. Let's see an example Article field configuration:

```
Article => {
    DescriptionShort => 'Please insert your comment here.',
    DescriptionLong => '',
    Display         => 1,
    Config          => {
        ArticleType  => 'note-internal',
        LabelSubject => '',
        LabelBody    => '',
    },
},
```

Let's look at the field configuration options:

2.4.3.7.1. DescriptionShort

Optional short description that is shown with the field title.

2.4.3.7.2. DescriptionLong

Optional longer field description that is shown then the mouse is over the field, for example advice on how to fill out the field.

2.4.3.7.3. Display

Controls if the field is shown and/or mandatory. Possible values:

- '0': field is invisible. This can be helpful if field values should automatically be set. The configured DefaultValue will be stored in this case.
- '1': field is visible, but optional.
- '2': field is visible and mandatory. The following fields can only be invisible or mandatory:

```
    QueueID
    Queue
    State
    StateID
    Lock
    LockID
    Priority
    PriorityID
    Type
    TypeID
```

If fields are configured as optional, and no value is submitted by the user, the Default Value will be saved when the Activity Dialog is submitted by the user.

2.4.3.7.4. DefaultValue

For fields with 'ID' (like QueueID, OwnerID), this refers to the database ID of the value. For other fields without 'ID' (like Queue, Owner), the DefaultValue must contain the value itself. Example:

```
Queue => {
    DescriptionShort => 'Queue',
    DescriptionLong  => 'Enter the queue here',
    Display          => 2,
    DefaultValue     => 'Raw',
},
```

2.4.3.8. FieldOrder

Here the display order of the fields is configured. IMPORTANT: Invisible fields also must be configured here, because only configured fields will be considered when saving. Fields which are not configured will not be saved.

2.4.3.9. SubmitAdviceText

Optional text to be shown right above the submit button for additional help or advice text.

2.4.3.10. SubmitButtonText

Optional custom text for the submit button.

2.4.4. Transition

A Transition decides - based on configurable conditions - which path in the Process is taken, i. e. to which Activity a Process ticket can be moved.

2.4.4.1. Transition configuration

Let's see an example:

```
$Self->{'Process::Transition'} = {
    'T1' => {
        Name => 'Transition 1',
        CreateTime => '14-03-2012 13:37:00', # optional
        CreateBy   => '1',                   # optional
        ChangeTime => '15-03-2012 13:37:00', # optional
        ChangeBy   => '15-03-2012 13:37:00', # optional
        Condition  => {
            Cond1  => {
                Fields => {
                    StateID => {
                        Type  => 'String',
                        Match => '1',
                    },
                },
            },
        },
    },
    'T2' => {
        Name       => 'Transition 2 optional',
        CreateTime => 'DATE',   # optional
        CreateBy   => 'USERID', # optional
        ChangeTime => 'DATE',   # optional
        ChangeBy => 'USERID',   # optional
        Condition => {
            Cond1 => {
                Queue               => 'Raw',
                DynamicField_Farbe  => '2',
                DynamicField_Anzahl => '1',
            },
        },
    },
};
```

2.4.4.2. Название

Name of the transition.

2.4.4.3. CreateTime

Time when it was created.

2.4.4.4. CreateBy

UID of the user who created this Transition.

2.4.4.5. ChangeTime

Last time when it was changed.

2.4.4.6. ChangeBy

UID of the last user who changed this Transition.

2.4.4.7. Condition

Contains all conditions that are neccessary for this Transition to take effect. Example:

```
Condition => {
    Type  => 'and',
    Cond1 => {
        Type   => 'and',
        Fields => {
            StateID => {
                Type  => 'String',
                Match => '1',
            },
            DynamicField_Marke => {
                Type  => 'String',
                Match => 'VW',
            },
        },
    },
    Cond2 => {
        Type => 'and',
        Fields => {
            Queue => {
                Type  => 'String',
                Match => 'Raw',
            },
        },
    },
},
```

Let's look at the condition configuration in detail.

2.4.4.7.1. Type (Condition)

Specifies the way the different condition elements are connected to each other. Possible values:

- 'and': This is the default. All conditions must be met for the transition to take effect.
- 'or': At least one condition must match.

- 'xor': Exactly one condition must match, not more.

2.4.4.7.2. Cond1

This is the name of an example condition. It can be freely chosen. Conditions are evaluated in sorted order.

2.4.4.7.3. Type (Cond)

Specifies the way how the individual field tests of this condition are connected to each other. Possible values:

- 'and': This is the default. All field tests must match for this condition to match.

- 'or': At least one field test must match.

- 'xor': Exactly one field test must match, not more.

2.4.4.7.4. Fields

Specifies the particular fields whose values should be tested. From our example:

```
Fields => {
    StateID => {
        Type  => 'String',
        Match => '1',
    },
```

2.4.4.7.5. StateID

Example of a field name. The following ticket fields can be used:

```
    Title
    State
    StateID
    Priority
    PriorityID
    Lock
    LockID
    Queue
    QueueID
    Customer
    CustomerID
    CustomerNo
    CustomerUserID
    Owner
    OwnerID
    Type
    TypeID
    SLA
    SLAID
    Service
    ServiceID
    Responsible
    ResponsibleID
    PendingTime
    DynamicField_$FieldName # for all DynamicFields
```

When testing a field with 'ID' (like SLAID), the database ID of the field will be used for testing, for other fields (like SLA) the actual value is used for testing.

2.4.4.7.6. Type

Determines the kind of field testing. Possible values:

- 'String': Compares the field value with the string specified in 'Match'. Matches if they are exactly the same.
- 'Hash': Compares the field value (hash) with the hash specified in 'Match'. All hash values must be the same.
- 'Array': Compares the field value (array) with the array specified in 'Match'. Both lists must be the same.
- 'Regex': The field value can be tested with a regular expression. It is important that 'Match' contains *qr{}xms* as a base condition. Between the braces the actual regular expression can be noted.
- 'Module': Allows you to use a perl module for condition checking. If it returns 1, the check was positive. You can find an example module in `Kernel/System/ProcessManagement/TransitionValidation/ValidateDemo.pm`.

2.4.5. Transition Actions

Transition Actions are actions which can be triggered after successfully applied transitions (when a process ticket moves from one activity to another). These Transition Actions can be used to perform different changes on the ticket, e. g. change the Queue or the Owner of the ticket, and you can also create your own Transition Actions to perform other complex changes.

2.4.5.1. Transition Action configuration

Let's see an example:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Queue Move',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet',
        Config => {
            Queue  => 'Junk',
            UserID => 123,
        },
    },
};
```

2.4.5.2. Название

The name of the Transition Action.

2.4.5.3. Module

Specifies the Perl module to be used.

2.4.5.4. Config

This parameter contains all settings which are required for the module. Its content depends on the particular Transition Action module which is used. Please see the documentation of the individual modules for details. In our example, only the Queue must be specified, Nevertheless we are also sending UserID parameter, by using the UserID parameter, the transition action will be executed impersonating the user with the given UserID.

The use of UserID inside the "Config" parameter of a Transition Action is accepted by all Transition Actions (since OTRS 3.2.4), in this example it could be particularly important if the user that triggers the Transition does not have permissions to move the ticket to the queue 'Junk', while the user with the UserID 123 might have.

2.4.5.5. Reusing Transition Action modules

To use Transition Action modules multiple times, just specify several TransitionActions in your configuration. Example:

```
    $Self->{'Process::TransitionAction'} = {
        'TA1' => {
            Name   => 'Queue Move Junk',
            Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet',
            Config => {
                Queue => 'Junk',
            },
        },
        'TA2' => {
            Name   => 'Queue Move Raw',
            Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet',
            Config => {
                Queue => 'Raw',
            },
        },
    };
```

Here the same module is used to move a process ticket into the 'Raw' queue, and another time to move it into the junk queue. The Transition Action which must be used for a particular Transition is determined from the 'Path' setting of the Process configuration.

2.4.5.6. Available Transition Actions

OTRS comes with several Transition Actions that can be used in your processes. Here you can find their documentation and how they need to be configured.

2.4.5.6.1. DynamicFieldSet

Sets one or more dynamic fields at a process ticket. Example:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Set DynamicField Master to Master and Approved to 1',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::DynamicFieldSet',
        Config => {
            MasterSlave => 'Master',
            Approved    => '1',

        },
    },
};
```

'Name' specifies the name of the configured TransitionAction.

'MasterSlave' and 'Approved' are given as examples of DynamicField names. The values of the fields ('Master' and '1') will be set by this TransitionAction.

2.4.5.6.2. TicketArticleCreate

Creates an article and can be used to create notes or email replies. Example:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Article Create Note Internal',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketArticleCreate',
        Config => {
            ArticleType      => 'note-internal',                                         # note-
external|phone|fax|sms|...
                                                                                         #
 excluding any email type
            SenderType       => 'agent',                                                 #
 agent|system|customer
            ContentType      => 'text/plain; charset=ISO-8859-15',                       # or
 optional Charset & MimeType
            Subject          => 'some short description',                                #
 required
            Body             => 'the message text',                                      #
 required
            HistoryType      => 'OwnerUpdate',                                           #
 EmailCustomer|Move|AddNote|PriorityUpdate|WebRequestCustomer|...
            HistoryComment   => 'Some free text!',
            From             => 'Some Agent <email@example.com>',                        # not
 required but useful
            To               => 'Some Customer A <customer-a@example.com>',              # not
 required but useful
            Cc               => 'Some Customer B <customer-b@example.com>',              # not
 required but useful
            ReplyTo          => 'Some Customer B <customer-b@example.com>',              # not
 required
            InReplyTo        => '<asdasdasd.12@example.com>',                            # not
 required but useful
            References       => '<asdasdasd.1@example.com> <asdasdasd.12@example.com>', # not
 required but useful
            NoAgentNotify    => 0,                                                       # if
 you don't want to send agent notifications
            AutoResponseType => 'auto reply',                                            # auto
 reject|auto follow up|auto reply/new ticket|auto remove

            ForceNotificationToUserID   => [ 1, 43, 56 ],                                # if
 you want to force somebody
            ExcludeNotificationToUserID => [ 43, 56 ],
                 # if you want full exclude somebody from notfications,
                 # will also be removed in To: line of article,
                 # higher prio as ForceNotificationToUserID
            ExcludeMuteNotificationToUserID => [ 43, 56 ],
                 # the same as ExcludeNotificationToUserID but only the
                 # sending gets muted, agent will still shown in To:
                 # line of article
        },
    },
};
```

'Name' specifies the name of the configured TransitionAction. It can be freely chosen, but should reflect the purpose of the configured action.

'ArticleType' defines the type of the article to be created. Possible values: phone, fax, sms, webrequest, note-internal, note-external and note-report.

SenderType defines the sender type of the article. Possible values: agent, system, customer.

'ContentType' defines the content type of the article. Possible values: 'text/plain; charset=ISO-8859-15' or any other valid charset and mime type.

'Subject' defines the article title. Mandatory.

'Body' defines the article content. Mandatory.

HistoryType defines the type of the history entry. Possible values: AddNote, ArchiveFlagUpdate, Bounce, CustomerUpdate, EmailAgent, EmailCustomer, EscalationResponseTimeNotifyBefore, EscalationResponseTimeStart, EscalationResponseTimeStop, EscalationSolutionTimeNotifyBefore, EscalationSolutionTimeStart, EscalationSolutionTimeStop, EscalationUpdateTimeNotifyBefore, EscalationUpdateTimeStart, EscalationUpdateTimeStop, FollowUp, Forward, Lock, LoopProtection, Merged, Misc, Move, NewTicket, OwnerUpdate, PhoneCallAgent, PhoneCallCustomer, PriorityUpdate, Remove, ResponsibleUpdate, SendAgentNotification, SendAnswer, SendAutoFollowUp, SendAutoReject, SendAutoReply, SendCustomerNotification, ServiceUpdate, SetPendingTime, SLAUpdate, StateUpdate, Subscribe, SystemRequest, TicketDynamicFieldUpdate, TicketLinkAdd, TicketLinkDelete, TimeAccounting, TypeUpdate, Unlock, Unsubscribe, WebRequestCustomer.

'HistoryComment' defines the content of the history entry.

'From', 'To', 'Cc' and 'ReplyTo' take email addresses in the notation specified above.

'InReplyTo' and 'References' take email message IDs.

'NoAgentNotify' - if set to 1, the email notification of the Agent will not be sent.

'AutoResponseType' can take the following values: auto follow up, auto reject, auto remove, auto reply, auto reply/new ticket.

'ForceNotificationToUserID', 'ExcludeNotificationToUserID', 'ExcludeMuteNotificationToUserID' can take a list of UserIDs that are either always notified, not notified or listed as notified but not actually sent a notification email.

2.4.5.6.3. TicketCustomerSet

Sets the customer of a process ticket. Example:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Customer Set Customer to test',
        Module => 'Kernel::System::Process::TransitionAction::TicketCustomerSet',
        Config => {
            No      => 'test',
            User    => 'client-user-123',
            # or in other words
            # CustomerID     => 'client123',
            # CustomerUserID => 'client-user-123',

        },
    },
};
```

'Name' specifies the name of the configured TransitionAction.

No or CustomerID set the Customer ID of the customer.

User or CustomerUserID set the Username of the customer.

2.4.5.6.4. TicketLockSet

Changes the lock of a process ticket. Example:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Set Lock to lock',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketLockSet',
        Config => {
            Lock   => 'lock',
            # or
            LockID => 2,
        },
    },
};
```

'Name' specifies the name of the configured TransitionAction.

'Lock' defines the new lock of the process ticket.

'LockID' defines the internal ID of the new lock.

2.4.5.6.5. TicketOwnerSet

Changes the owner of a process ticket. Example:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Owner Set root@localhost',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketOwnerSet',
        Config => {
            Owner => 'root@localhost',
            # or
            OwnerID => 1,
        },
    },
};
```

'Name' specifies the name of the configured TransitionAction.

'Owner' specifies the login name of the new owner.

'OwnerID' specifies the internal ID of the new owner.

2.4.5.6.6. TicketQueueSet

Moves the ticket into a target queue. Example:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Queue Move Raw',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet',
        Config => {
            Queue => 'Raw',
            # or
            # QueueID => '2',
        },
    },
};
```

'Name' specifies the name of the configured TransitionAction.

'Queue' specifies the name of the target queue.

'QueueID' specifies the internal ID of the target queue.

2.4.5.6.7. TicketResponsibleSet

Changes the responsible of a process ticket. Example:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Responsible Set root@localhost',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketResponsibleSet',
        Config => {
            Responsible => 'root@localhost',
            # or
            ResponsibleID => 1,
        },
    },
};
```

'Name' specifies the name of the configured TransitionAction.

'Responsible' specifies the login name of the new responsible.

'ResponsibleID' specifies the internal ID of the new responsible.

2.4.5.6.8. TicketServiceSet

Assigns a service to a process ticket. The ticket requires to have a customer and the service must be assigned to that customer. Example:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Set MyService service',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketServiceSet',
        Config => {
            Service   => 'MyService',
            # or
            ServiceID => 123,
        },
    },
};
```

'Name' specifies the name of the configured TransitionAction.

'Service' defines the new service of the process ticket. The full name is required (e.g. GramdFatherService::FatherService::SonService).

'ServiceID' defines the internal ID of the new service.

2.4.5.6.9. TicketSLASet

Assigns a service level agreement to a process ticket. The ticket requires to have a service and the SLA must be assigned to that service. Example:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Set MySLA SLA',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketSLASet',
        Config => {
            SLA   => 'MyService',
            # or
            SLAID => 123,
        },
    },
};
```

'Name' specifies the name of the configured TransitionAction.

'SLA' defines the new service level agreement of the process ticket.

'SLAID' defines the internal ID of the new SLA.

2.4.5.6.10. TicketStateSet

Changes the state of a process ticket. Example:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Set State to open',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketStateSet',
        Config => {
            State   => 'open',
            # or
            StateID => 4,

            PendingTimeDiff => 123,
        },
    },
};
```

'Name' specifies the name of the configured TransitionAction.

'State' defines the new state of the process ticket.

'StateID' defines the internal ID of the new state.

'PendingTimeDiff' used only for pending type states, defines the time difference in seconds relative (relative to the Transition Action execution time) to set ticket pending time (e.g. 3600 means that the pending time is 1hr after the Transition Action is executed).

2.4.5.6.11. TicketTitleSet

Sets the ticket title of a process ticket. Example:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Set Ticket Title to Ticket-title',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketTitleSet',
        Config => {
            Title => 'Ticket-title',
        },
    },
};
```

'Name' specifies the name of the configured TransitionAction.

'Title' specifies the new title of the ticket.

2.4.5.6.12. TicketTypeSet

Sets the ticket type of a process ticket. Example:

```
$Self->{'Process::TransitionAction'} = {
    'TA1' => {
        Name   => 'Set Ticket Type to default',
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketTypeSet',
        Config => {
            Type     => 'default',
            # or
            # TypeID => '1',
        },
    },
};
```

'Name' specifies the name of the configured TransitionAction.

'Type' specifies the name of the ticket type.

'TypeID' specifies the internal ID of the ticket type.

2.4.6. Списки Контроля Доступа (ACLs)

With the help of ACLs, you can limit selectable values in process tickets. Please also see the ACL reference for a description of the full ticket ACL syntax.

2.4.6.1. ACL configuration

ACLs can only be defined in `Kernel/Config.pm`. Example:

```
$Self->{TicketAcl}->{'001-ACL-ProcessProperties'} = {
    Properties => {
        Process => {
            ProcessEntityID        => ['P1'],
            ActivityEntityID       => ['A1'],
            ActivityDialogEntityID => ['AD1'],
        }
    },
    Possible => {
        ActivityDialog => ['AD1', 'AD3'],
    },
    PossibleNot => {
        ActivityDialog => ['AD3'],
    },
};
```

2.4.6.2. 001-ACL-ProcessProperties

Name of the ACL rule. For further information on ACL rules in general, please consult the ACL manual.

2.4.6.3. Process

This is the section that is used to check if an ACL must be applied. If it has the specified values, the rule is applied. The following values can be used:

2.4.6.3.1. ProcessEntityID

The ID of a process that the process. Matches if the ticket is assigned to this process.

2.4.6.3.2. ActivityEntityID

The ID of the Activity that the process ticket currently is assigned to.

2.4.6.3.3. ActivityDialogEntityID

The ID of the Activity Dialog that is currently open for a process ticket.

2.4.6.4. Possible/PossibleNot Activity Dialog

Here you can specify a list of Activity Dialog IDs. This list will limit the possible Activity Dialogs that are offered to the user in the ticket zoom mask.

'Possible' lists the Activity Dialogs that are allowed. The setting above will only allow 'AD1' and 'AD3' of the list of configured Activity Dialogs.

'PossibleNot' lists the Activity Dialogs that are not allowed. In the example above, the setting will remove 'AD3' from the list of configured Activity Dialogs.

If both 'Possible' and 'PossibleNot' are specified, the list of configured Activity Dialogs will first be filtered by 'Possible', leaving only 'AD1' and 'AD3' in our example. Then 'PossibleNot' will be applied and filter out 'AD3', so that only 'AD1' remains and is shown as a possible Activity Dialog that the user can use.

If multiple ACL rules match, the intersection of all matching rules will be calculated to determine the possible Activity Dialogs. Example:

Configured Activity Dialogs: 'AD1', 'AD2', 'AD3', 'AD4', 'AD5', 'AD6', 'AD7'.

```
$Self->{TicketAcl}->{'001-ACL-Status'} = {
    Properties => {
        Ticket => {
            Status => 'new',
        }
    },
    Possible => {
        ActivityDialog => ['AD1', 'AD2', 'AD3', 'AD6', 'AD7'],
    },
};
$Self->{TicketAcl}->{'002-ACL-Queue'} = {
    Properties => {
        Ticket => {
            Queue => ['Raw']
        }
    },
    Possible => {
        ActivityDialog => ['AD2', 'AD3', 'AD4', 'AD7'],
    },
};
$Self->{TicketAcl}->{'003-ACL-Priority'} = {
    Properties => {
        Ticket => {
            Priority => ['3 normal']
        }
    },
    PossibleNot => {
        ActivityDialog => ['AD3', 'AD4'],
    },
};
```

If a process ticket has the state 'new', is in the 'Raw' queue and has a priority '3 normal', then all three ACL rules will match.

The first rule reduces the Activity Dialogs from 'AD1', 'AD2', 'AD3', 'AD4', 'AD5', 'AD6', 'AD7' to 'AD1', 'AD2', 'AD3', 'AD6', 'AD7' and forbids 'AD4' and 'AD5'.

The second rule will now further reduce the remaining Activity Dialogs. In our example, 'AD2', 'AD3', 'AD7' will remain.

Now the third rule will further reduce the list by 'PossibleNot'. 'AD3' is removed from the list. 'AD4' is not removed, since it was not on the list in the first place. At the end, 'AD2' and 'AD7' remain as possible Activity Dialogs that the user can utilize.

It is also possible to limit the processes that can be displayed in the "New process ticket" screen, the functionality is similar to limiting the ActivityDialogs with one exception: The ACLs could only be based on Users.

See examples below:

```
$Self->{TicketAcl}->{'200-ACL-Process'} = {
    # match properties
    Properties => {
        User => {
            UserID => [2, 3],
        },
    },
    Possible => {
       Process => ['P1', 'P2', 'P3'],
    },
    PossibleNot => {
        Process => ['P4'],
    },
};
```

```
$Self->{TicketAcl}->{'201-ACL-Process'} = {
    # match properties
    Properties => {
        User => {
            Group_rw => [ 'MyGroup' ],
        },
    },
    Possible => {
       Process => ['P1', 'P2', 'P3'],
    },
    PossibleNot => {
        Process => ['P4'],
    },
};
```

```
$Self->{TicketAcl}->{'202-ACL-Process'} = {
    # match properties
    Properties => {
        User => {
            Role => [ 'MyRole' ],
        },
    },
    Possible => {
       Process => ['P1', 'P2', 'P3'],
    },
    PossibleNot => {
        Process => ['P4'],
    },
};
```

3. Создание своих собственных тем (шаблонов)

Для OTRS можно создавать свои собственные темы и использовать в качестве веб-интерфейса тот шаблон, который вам нравится больше всего. Для создания собственной темы нужно настроить шаблоны вывода под свои нужды.

More information on the syntax and structure of output templates can be found in the Developer Manual at http://otrs.github.io/doc, especialy in the chapter on *templates*.

В качестве примера, выполните следующие шаги для создания новой темы, которая называется "Компания":

1. Создайте директорию и назовите ее `Kernel/Output/HTML/Company` и скопируйте все файлы, которые хотите изменить, из `Kernel/Output/HTML/Standard` в новую директорию.

   ### Важно

   Копируйте только те файлы, которые вы действительно планируете изменять. OTRS автоматически получит недостающие файлы из Стандартной темы. На более позних этапах такой подход позволит сделать модернизацию более простой.

2. Настройте файлы в директории `Kernel/Output/HTML/Company` и изментите шаблоны по своему вкусу.

3. Чтобы активировать новую тему, перейдите по ссылке Frontend::Themes и добавьте ее в SysConfig.

Теперь новая тема должна быть доступна. Вы можете выбрать эту тему перейдя по ссылке предпочтения.

Предупреждение

Не изменяйте файлы темы, поставляемой с OTRS, поскольку эти изменения будут утеряны после обновления. Создавайте свои собственные темы только путем выполнения шагов, описанных выше.

4. Локализайия интерфейса OTRS

OTRS предлагает многоязычную поддержку для своего веб-интерфейса.

Procedures for localization for the OTRS framework, steps to be followed to create a new language translation, as well as procedures for translation customizations, can be found in the "Language Translations" chapter from the developer manual on http://otrs.github.io/doc.

Глава 6. Настройка Производительности

Presented below is a list of performance enhancing techniques for your OTRS installation, including configuration, coding, memory use, and more.

1. OTRS

There are several options for improving OTRS performance.

1.1. TicketIndexModule

Есть два типа хранилища информации для хранения индексов заявок:

- Kernel::System::Ticket::IndexAccelerator::RuntimeDB (по умолчанию), "на лету" генерирует из таблицы заявок шаблон просмотра для каждой очереди. Пока в системе не больше 60, 000 открытых заявок, проблем с производительностью не будет.

- Kernel::System::Ticket::IndexAccelerator::StaticDB - один из самых мощных модулей, его следует использовать, когда у вас еть больше 80.000 открытых заявок. Он использует дополнительную таблицу ticket_index, которая работает как обычный шаблон. Используйте `bin/otrs.RebuildTicketIndex.pl` для получения начального индекса после создания хранилища информации.

Вы можете изменить IndexAccelerator через SysConfig.

1.2. TicketStorageModule

Существует два различных хранилица для хранения заявок/статей:

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

  Pro: Если пользователь вашего веб-сервера не является пользователем 'otrs', используйте этот модуль, чтобы избежать проблем с правами доступа.

  Противопоказания: Не совсем целесообразно хранить вложения в базе данных. Будьте осторожны, при храненении больших объектов. Настройте конфигурационный параметр MySQL "set-variable=max_allowed_packet=8M" для хранения объектов размером 8 MB (по умолчанию это 2M).

- Configure Kernel::System::Ticket::ArticleStorageFS для хранения вложений и т.д. в локальной файловой системе. Примечание: Рекомендовано при больших установках.

  Pro: Это быстро!

  Con: Your web server user should be the 'otrs' user. Also, if you have multiple front-end servers, you should make sure the filesystem is shared between the servers. Place it on an NFS share or preferably a SAN or similar solution.

Note: you can switch from one back-end to the other on the fly. You can switch the backend in the SysConfig, and then run the command line utility `otrs.ArticleStorageSwitch.pl` to put the articles from the database onto the filesystem or the other way around. You can use the *-s* and *-d* options to specify the source and destination back-ends. Please note that the entire process can take considerable time to run, depending on the number of articles you have and the available CPU power and/or network capacity.

```
shell> bin/otrs.ArticleStorageSwitch.pl -s ArticleStorageDB -d ArticleStorageFS
```

Сценарий Переключение хранилища данных с базы данных на файловую систему.

1.3. Архивирование Заявок

Поскольку OTRS может использоваться в качестве системы аудита доказательств, то удаление закрытых заявко не очень хорошая идея. Именно по этому мы реализовали функцию архивирования заявок.

Tickets that match certain criteria can be marked as "archived" These tickets are not accessed if you do a regular ticket search or run a Generic Agent job. The system itself does not have to deal with a huge amount of tickets any longer as only the "latest" tickets are taken into consideration when using OTRS. This can result in a huge performance gain on large systems.

Для использования функции архивации выполните следующие действия:

1. Включение архивирование системы в SysConfig

   В Панели Администрирования перейдите в SysConfig и выберите группу `Заявка`. В `Core::Ticket` найдите опцию `Ticket::ArchiveSystem`, по умолчанию установленную в значение "нет". Измените значение этой настройки на "да" и сохраните изменения.

2. Определение работы GenericAgent

   В Панели Администратора выберите GenericAgent и добавьте новое задание (работу).

   a. Настройки Задания

      Введите имя для работы архивирования, и выберите надлежащие опции для планирования этой работы.

   b. Ticket Filter

      Фильтр заявок производит поиск заявок, которые отвечают выбраным критериям. Хорошей идеей будет заархивировать закрытые заявки, которые были закрыты за несколько месяцев до этого.

   c. Действия над заявками

      В этой части установите поле "Архивировать выбранные заявки" в "архив заявок".

   d. Сохранить работу

      В конце страницы у вас будет возможность сохранить работу (задание).

   e. Обработанные заявки

      Система отобразит все заявки, которые будут заархивированы при выполнении задания Generic Agent-ом.

3. Поиск Заявок

   При поиске заявок, система по умолчанию производит поиск среди не архивированных заявок. Установите критерий поиска "поиск в архивах", если нужно чтобы поиск происходил также и в архивированых заявках.

2. База данных

Решения зависят от используемой базы данных. Изучайте документацию к используемой базе данных или же обратитесь за помощью к администратору.

2.1. MySQL

If you use the 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 must defragment the datafile (tables) with the "optimize" command.

Нужно попытаться сделать это, если сервер mysqld использует много процессорного времени. Оптимизация таблиц ticket_history и article (см Сценарий ниже).

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

Сценарий: Оптимизация таблиц базы данных.

2.2. PostgreSQL

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

- http://www.revsys.com/writings/postgresql-performance.html
- http://varlena.com/GeneralBits/Tidbits/perf.html
- http://varlena.com/GeneralBits/Tidbits/annotated_conf_e.html

Если производительность остается на неудовлетворительном уровне, мы предлагаем вам присоединится к спискам рассылки "PostgreSQL Performance" (http://www.postgresql.org/community/lists/) и задавать вопросы именно там. Люди в этом списке рассылке очень дружелюбны а всегада постараются вам помочь.

3. Веб-сервер

Конечно же нужно использовать mod_perl 2.0 (http://perl.apache.org/). Он быстрее (~ * 100) чем "чистый" cgi. Но он требует больше оперативной памяти (RAM).

3.1. Предустановленное соединение с базой данных

При запуске веб-сервера можно иметь заранее установленное соединение с базой данных. Такой подход существенно сэкономит время (см. README.webserver).

3.2. Предварительно загруженные модули - startup.pl

Чтобы все работало быстрее и занимало меньше памяти используйте сценарий запуска `scripts/apache2-perl-startup.pl` для предварительной загрузки/компиляции Perl модулей на сервере с mod_perl (см. README.webserver).

3.3. Перезагрузка Perl-модулей во время обновления с диска

By default Apache::Reload is used in scripts/apache2-httpd.include.conf. Disable it and you will get 8% more speed. But remember to restart the web server if you install any modules via the OTRS Package Manager, or any values in your SysConfig or in Kernel/Config.pm.

Important: this would also mean you can't use the OTRS Package Manager via the web interface, you need to use the command line variant - `bin/otrs.PackageManager.pl`.

3.4. Выбор Правильной Стратегии

If you have a larger installation, e.g. over 1,000 new tickets per day and over 40 agents, it is a good idea to read the chapters on Performance of the mod_perl User's Guide (http://perl.apache.org/docs/2.0/user/index.html).

3.5. mod_gzip/mod_deflate

Используйте mod_deflate для Apache2, если у вас "слабая" полоса пропускания . При загрузке html-страницы размером в 45к, mod_gzip/mod_deflate сожмет ее и страница будет занимать около 7к. Недостатком такого подхода является дополнительная нагрузка на сервер.

Приложение A. Дополнительные ресурсы

We try to support you with the very latest information about OTRS. We also give you an opportunity to provide us with your feedback.

1. Website OTRS Group

You can find the website of the OTRS Group, the company behind OTRS, at *http://www.otrs.com*. It contains a lot of product-related information, such as white papers, release notes, case studies and so on.

2. Списки рассылки

The Table A-1 displays our various community mailing lists.

Таблица A.1. Списки рассылки

Название	**Описание**	**Домашняя страница**
announce@otrs.org	Низкий трафик, на английском языке, для объявлений о новых релизах OTRS и решениях о безопасности.	http://lists.otrs.org/cgi-bin/listinfo/announce
otrs@otrs.org	Medium to high traffic list, in English, where you can find all sorts of relevant questions and support for the product.	http://lists.otrs.org/cgi-bin/listinfo/otrs
otrs-de@otrs.org	Medium to high traffic list, in German, where you can find all sorts of relevant questions and support for the product.	http://lists.otrs.org/cgi-bin/listinfo/otrs-de
dev@otrs.org	Средний трафик на английском языке, где OTRS-разработчики обсуждают различные планы и вопросы реализации.	http://lists.otrs.org/cgi-bin/listinfo/dev
i18n@otrs.org	Небольшое количество пользователей на английском языке с вопросами интернационализации и локализации. Если вы есть или только хотите стать переводчиком проекта OTRS или имеете проблемы с одним из наших приложений, это именно то место, куда вам нужно.	http://lists.otrs.org/cgi-bin/listinfo/i18n

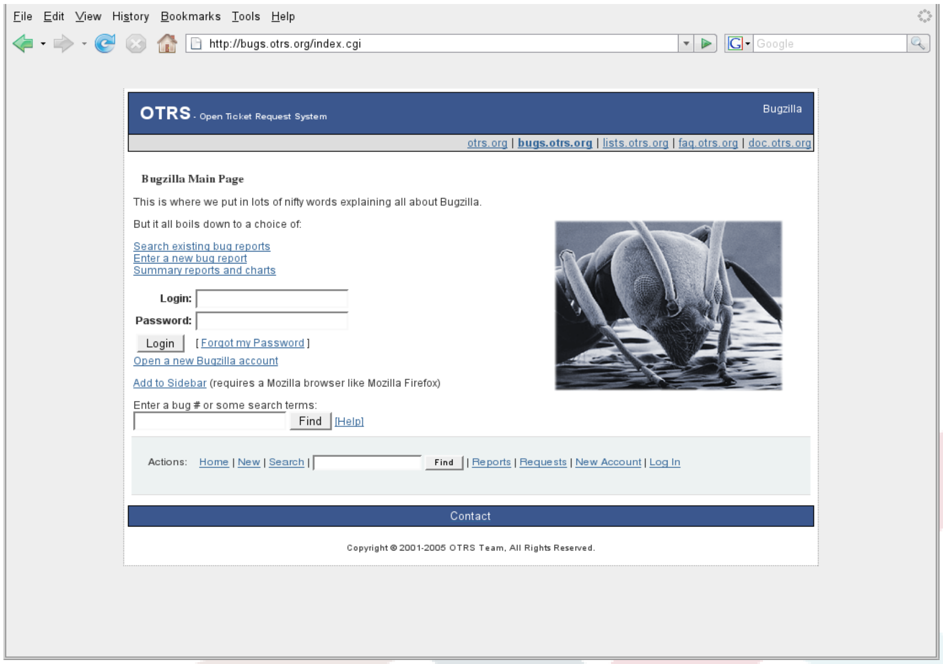
To subscribe to any of these lists, visit the following link: http://lists.otrs.org/.

3. User Forums

You can find community user forums at *http://forums.otrs.org*. It allows you to get in contact with users all around the world and exchange experiences regarding the use of OTRS.

4. Трекинг ошибок

To submit bugs visit http://bugs.otrs.org/ (see Figure below). Please take note of the difference between a bug and a configuration issue. Configuration issues are problems that you encounter when setting a system, or general questions regarding 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.



http://bugs.otrs.org/index.cgi

otrs.org | **bugs.otrs.org** | lists.otrs.org | faq.otrs.org | doc.otrs.org

Bugzilla Main Page

This is where we put in lots of nifty words explaining all about Bugzilla.

But it all boils down to a choice of:

Search existing bug reports
Enter a new bug report
Summary reports and charts

Login:
Password:

[Forgot my Password]
Open a new Bugzilla account

Add to Sidebar (requires a Mozilla browser like Mozilla Firefox)

Enter a bug # or some search terms:
[Help]

Actions: Home | New | Search | | Reports | Requests | New Account | Log In

Copyright © 2001-2005 OTRS Team, All Rights Reserved.

Рисунок: Трекинг ошибок.

For configuration issues, you should either use the *commercial support, available from OTRS.com*, or the public mailing lists.

Вы поможете нам улучшить продукт если сообщите об ошибке. Мы ценним ваш вклад!

5. Коммерческая поддержка

For services (support, consulting, development, and training) you can contact the company behind OTRS, OTRS AG. Our offices are located in Germany, USA, Mexico, the Netherlands, and in other countries. Visit our website for contact information: http://www.otrs.com/en/corporate-navigation/contact/

Приложение B. Configuration Options Reference

1. DynamicFields

1.1. DynamicFields::Driver::Registration

1.1.1. DynamicFields::Driver###Text

Description:	DynamicField backend registration.
Group:	DynamicFields
SubGroup:	DynamicFields::Driver::Registration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DynamicFields::Driver'}->{'Text'} =  {`
	`  'ConfigDialog' => 'AdminDynamicFieldText',`
	`  'DisplayName' => 'Text',`
	`  'Module' => 'Kernel::System::DynamicField::Driver::Text'`
	`};`

1.1.2. DynamicFields::Driver###TextArea

Description:	DynamicField backend registration.
Group:	DynamicFields
SubGroup:	DynamicFields::Driver::Registration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DynamicFields::Driver'}->{'TextArea'} =  {`
	`  'ConfigDialog' => 'AdminDynamicFieldText',`
	`  'DisplayName' => 'Textarea',`
	`  'Module' => 'Kernel::System::DynamicField::Driver::TextArea'`
	`};`

1.1.3. DynamicFields::Driver###Checkbox

Description:	DynamicField backend registration.
Group:	DynamicFields
SubGroup:	DynamicFields::Driver::Registration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DynamicFields::Driver'}->{'Checkbox'} =  {`
	`  'ConfigDialog' => 'AdminDynamicFieldCheckbox',`
	`  'DisplayName' => 'Checkbox',`
	`  'Module' => 'Kernel::System::DynamicField::Driver::Checkbox'`
	`};`

1.1.4. DynamicFields::Driver###Dropdown

Description:	DynamicField backend registration.
Group:	DynamicFields
SubGroup:	DynamicFields::Driver::Registration
Valid:	1

Required:	0

Config-Setting:	`$Self->{'DynamicFields::Driver'}->{'Dropdown'} =  {`
	`  'ConfigDialog' => 'AdminDynamicFieldDropdown',`
	`  'DisplayName' => 'Dropdown',`
	`  'Module' => 'Kernel::System::DynamicField::Driver::Dropdown'`
	`};`

1.1.5. DynamicFields::Driver###DateTime

Description:	DynamicField backend registration.
Group:	DynamicFields
SubGroup:	DynamicFields::Driver::Registration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DynamicFields::Driver'}->{'DateTime'} =  {`
	`  'ConfigDialog' => 'AdminDynamicFieldDateTime',`
	`  'DisplayName' => 'Date / Time',`
	`  'Module' => 'Kernel::System::DynamicField::Driver::DateTime'`
	`};`

1.1.6. DynamicFields::Driver###Date

Description:	DynamicField backend registration.
Group:	DynamicFields
SubGroup:	DynamicFields::Driver::Registration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DynamicFields::Driver'}->{'Date'} =  {`
	`  'ConfigDialog' => 'AdminDynamicFieldDateTime',`
	`  'DisplayName' => 'Date',`
	`  'Module' => 'Kernel::System::DynamicField::Driver::Date'`
	`};`

1.1.7. DynamicFields::Driver###Multiselect

Description:	DynamicField backend registration.
Group:	DynamicFields
SubGroup:	DynamicFields::Driver::Registration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DynamicFields::Driver'}->{'Multiselect'} =  {`
	`  'ConfigDialog' => 'AdminDynamicFieldMultiselect',`
	`  'DisplayName' => 'Multiselect',`
	`  'ItemSeparator' => ', ',`
	`  'Module' => 'Kernel::System::DynamicField::Driver::Multiselect'`
	`};`

1.2. DynamicFields::ObjectType::Registration

1.2.1. DynamicFields::ObjectType###Ticket

Description:	DynamicField object registration.

Group:	DynamicFields
SubGroup:	DynamicFields::ObjectType::Registration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DynamicFields::ObjectType'}->{'Ticket'} =  {`
	`  'DisplayName' => 'Ticket',`
	`  'Module' => 'Kernel::System::DynamicField::ObjectType::Ticket'`
	`};`

1.2.2. DynamicFields::ObjectType###Article

Description:	DynamicField object registration.
Group:	DynamicFields
SubGroup:	DynamicFields::ObjectType::Registration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DynamicFields::ObjectType'}->{'Article'} =  {`
	`  'DisplayName' => 'Article',`
	`  'Module' => 'Kernel::System::DynamicField::ObjectType::Article'`
	`};`

1.3. Frontend::Admin::ModuleRegistration

1.3.1. Frontend::Module###AdminDynamicField

Description:	Frontend module registration for the agent interface.
Group:	DynamicFields
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminDynamicField'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'Loader' => {`
	`    'CSS' => [`
	`      'Core.Agent.Admin.DynamicField.css'`
	`    ],`
	`    'JavaScript' => [`
	`      'Core.Agent.Admin.DynamicField.js'`
	`    ]`
	`  },`
	`  'NavBarModule' => {`
	`    'Block' => 'Ticket',`
	`    'Description' => 'Create and manage dynamic fields.',`
	`    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`    'Name' => 'Dynamic Fields',`
	`    'Prio' => '1000'`
	`  },`
	`  'NavBarName' => 'Admin',`
	`  'Title' => 'Dynamic Fields GUI'`
	`};`

1.3.2. Frontend::Module###AdminDynamicFieldText

Description:	Frontend module registration for the agent interface.
Group:	DynamicFields
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminDynamicFieldText'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.DynamicField.js',
      'Core.Agent.Admin.DynamicFieldText.js'
    ]
  },
  'Title' => 'Dynamic Fields Text Backend GUI'
};
```

1.3.3. Frontend::Module###AdminDynamicFieldCheckbox

Description:	Frontend module registration for the agent interface.
Group:	DynamicFields
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminDynamicFieldCheckbox'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.DynamicField.js'
    ]
  },
  'Title' => 'Dynamic Fields Checkbox Backend GUI'
};
```

1.3.4. Frontend::Module###AdminDynamicFieldDropdown

Description:	Frontend module registration for the agent interface.
Group:	DynamicFields
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminDynamicFieldDropdown'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.DynamicField.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.DynamicField.js',
      'Core.Agent.Admin.DynamicFieldDropdown.js'
    ]
  },
  'Title' => 'Dynamic Fields Drop-down Backend GUI'
};
```

1.3.5. Frontend::Module###AdminDynamicFieldDateTime

Description:	Frontend module registration for the agent interface.
Group:	DynamicFields
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
 $Self->{'Frontend::Module'}->{'AdminDynamicFieldDateTime'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.DynamicField.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.DynamicField.js',
      'Core.Agent.Admin.DynamicFieldDateTime.js'
    ]
  },
  'Title' => 'Dynamic Fields Date Time Backend GUI'
};
```

1.3.6. Frontend::Module###AdminDynamicFieldMultiselect

Description:	Frontend module registration for the agent interface.
Group:	DynamicFields
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
 $Self->{'Frontend::Module'}->{'AdminDynamicFieldMultiselect'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.DynamicField.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.DynamicField.js',
      'Core.Agent.Admin.DynamicFieldMultiselect.js'
    ]
  },
  'Title' => 'Dynamic Fields Multiselect Backend GUI'
};
```

1.4. Frontend::Agent::Preferences

1.4.1. PreferencesGroups###DynamicField

Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	DynamicFields
SubGroup:	Frontend::Agent::Preferences
Valid:	0
Required:	0

Config-Setting:	$Self->{'PreferencesGroups'}->{'DynamicField'} = {
	  'Active' => '1',
	  'Block' => 'Input',
	  'Column' => 'Other Settings',
	  'Data' => '[% Env("UserDynamicField_NameX") %]',
	  'Key' => 'Default value for NameX',
	  'Label' => 'NameX',
	  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	  'PrefKey' => 'UserDynamicField_NameX',
	  'Prio' => '7000'
	};

1.4.2. PreferencesGroups###DynamicFieldsOverviewPageShown

Description:	Parameters for the pages (in which the dynamic fields are shown) of the dynamic fields overview.
Group:	DynamicFields
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	$Self->{'PreferencesGroups'}->{'DynamicFieldsOverviewPageShown'} = {
	  'Active' => '0',
	  'Column' => 'Other Settings',
	  'Data' => {
	    '10' => '10',
	    '15' => '15',
	    '20' => '20',
	    '25' => '25',
	    '30' => '30',
	    '35' => '35'
	  },
	  'DataSelected' => '25',
	  'Key' => 'Dynamic fields limit per page for Dynamic Fields Overview',
	  'Label' => 'Dynamic Fields Overview Limit',
	  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	  'PrefKey' => 'AdminDynamicFieldsOverviewPageShown',
	  'Prio' => '8000'
	};

2. Framework

2.1. Core

2.1.1. SecureMode

Description:	Disables the web installer (http://yourhost.example.com/otrs/installer.pl), to prevent the system from being hijacked. If set to "No",

	the system can be reinstalled and the current basic configuration will be used to pre-populate the questions within the installer script. If not active, it also disables the GenericAgent, PackageManager and SQL Box.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SecureMode'} =  '0';`

2.1.2. Frontend::DebugMode

Description:	Enables or disable the debug mode over frontend interface.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::DebugMode'} =  '0';`

2.1.3. ConfigLevel

Description:	Sets the configuration level of the administrator. Depending on the config level, some sysconfig options will be not shown. The config levels are in in ascending order: Expert, Advanced, Beginner. The higher the config level is (e.g. Beginner is the highest), the less likely is it that the user can accidentally configure the system in a way that it is not usable any more.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'ConfigLevel'} =  '100';`

2.1.4. ProductName

Description:	Defines the name of the application, shown in the web interface, tabs and title bar of the web browser.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'ProductName'} =  'OTRS';`

2.1.5. SystemID

Description:	Defines the system identifier. Every ticket number and http session string contains this ID. This ensures that only tickets which belong

	to your system will be processed as follow-ups (useful when communicating between two instances of OTRS).
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SystemID'} =  '10';`

2.1.6. FQDN

Description:	Defines the fully qualified domain name of the system. This setting is used as a variable, OTRS_CONFIG_FQDN which is found in all forms of messaging used by the application, to build links to the tickets within your system.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'FQDN'} =  'yourhost.example.com';`

2.1.7. HttpType

Description:	Defines the type of protocol, used by the web server, to serve the application. If https protocol will be used instead of plain http, it must be specified here. Since this has no affect on the web server's settings or behavior, it will not change the method of access to the application and, if it is wrong, it will not prevent you from logging into the application. This setting is only 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:	`$Self->{'HttpType'} =  'http';`

2.1.8. ScriptAlias

Description:	Sets the prefix to the scripts folder on the server, as configured on the web server. This setting is used as a variable, OTRS_CONFIG_ScriptAlias which is found in all forms of messaging used by the application, to build links to the tickets within the system.
Group:	Framework
SubGroup:	Core
Valid:	1

Required:	1
Config-Setting:	`$Self->{'ScriptAlias'} =  'otrs/';`

2.1.9. AdminEmail

Description:	Defines the system administrator's email address. It will be displayed in the error screens of the application.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'AdminEmail'} =  'admin@example.com';`

2.1.10. Organization

Description:	Company name which will be included in outgoing emails as an X-Header.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Organization'} =  'Example Company';`

2.1.11. DefaultLanguage

Description:	Defines the default front-end language. All the possible values are determined by the available language files on the system (see the next setting).
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'DefaultLanguage'} =  'en';`

2.1.12. DefaultUsedLanguages

Description:	Defines all the languages that are available to the application. The Key/Content pair links the front-end display name to the appropriate language PM file. The "Key" value should be the base-name of the PM file (i.e. de.pm is the file, then de is the "Key" value). The "Content" value should be the display name for the front-end. Specify any own-defined language here (see the developer documentation http://doc.otrs.org/ for more infomation). Please remember to use the HTML equivalents for non-ASCII characters (i.e. for the German oe = o umlaut, it is necessary to use the &ouml; symbol).

<table>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Core</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>1</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'DefaultUsedLanguages'} = {
  'ar_SA' => 'Arabic (Saudi Arabia)',
  'bg' => 'Bulgarian
 (&#x0411;&#x044a;&#x043b;&#x0433;&#x0430;&#x0440;&#x0441;&#x043a;&#x0438;)',
  'ca' => 'Catal&agrave;',
  'cs' => 'Czech (&#x010c;esky)',
  'da' => 'Dansk',
  'de' => 'Deutsch',
  'el' => 'Greek
 (&#x0395;&#x03bb;&#x03bb;&#x03b7;&#x03bd;&#x03b9;&#x03ba;&#x03ac;)',
  'en' => 'English (United States)',
  'en_CA' => 'English (Canada)',
  'en_GB' => 'English (United Kingdom)',
  'es' => 'Espa&ntilde;ol',
  'es_CO' => 'Espa&ntilde;ol (Colombia)',
  'es_MX' => 'Espa&ntilde;ol (M&eacute;xico)',
  'et' => 'Eesti',
  'fa' => 'Persian (&#x0641;&#x0627;&#x0631;&#x0633;&#x0649;)',
  'fi' => 'Suomi',
  'fr' => 'Fran&ccedil;ais',
  'fr_CA' => 'Fran&ccedil;ais (Canada)',
  'he' => "Hebrew
 (\x{5e2}\x{5b4}\x{5d1}\x{5b0}\x{5e8}\x{5b4}\x{5d9}\x{5ea})",
  'hi' => 'Hindi',
  'hr' => 'Hrvatski',
  'hu' => 'Magyar',
  'it' => 'Italiano',
  'ja' => 'Japanese (&#x65e5;&#x672c;&#x8a9e)',
  'lt' => "Lietuvi\x{173} kalba",
  'lv' => 'Latvijas',
  'ms' => 'Malay',
  '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)',
  'sl' => "Slovenian (Sloven\x{161}\x{10d}ina)",
  'sr_Cyrl' => "Serbian Cyrillic
 (\x{441}\x{440}\x{43f}\x{441}\x{43a}\x{438})",
  'sr_Latn' => 'Serbian Latin (Srpski)',
  'sv' => 'Svenska',
  'tr' => 'T&uuml;rk&ccedil;e',
  'uk' => 'Ukrainian
 (&#x0423;&#x043a;&#x0440;&#x0430;&#x0457;&#x043d;&#x0441;&#x044c;&#x043a;&#x0430;)',
  'vi_VN' => 'Vietnam (Vi&#x0246;t Nam)',
  'zh_CN' => 'Chinese (Sim.) (&#x7b80;&#x4f53;&#x4e2d;&#x6587;)',
  'zh_TW' => 'Chinese (Tradi.) (&#x6b63;&#x9ad4;&#x4e2d;&#x6587;)'
};
</pre></td></tr>
</table>

2.1.13. DefaultTheme

Description:	Defines the default front-end (HTML) theme to be used by the agents and customers. If you like, you can add your own theme. Please refer the administrator manual located at http://doc.otrs.org/.
Group:	Framework

SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'DefaultTheme'} =  'Standard';`

2.1.14. DefaultTheme::HostBased

Description:	It is possible to configure different themes, for example to distinguish between agents and customers, to be used on a per-domain basis within the application. Using a regular expression (regex), you can configure a Key/Content pair to match a domain. The value in "Key" should match the domain, and the value in "Content" should be a valid theme on your system. Please see the example entries for the proper form of the regex.
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0

Config-Setting:	`$Self->{'DefaultTheme::HostBased'} =  {`
	`  'host1\\.example\\.com' => 'SomeTheme1',`
	`  'host2\\.example\\.com' => 'SomeTheme2'`
	`};`

2.1.15. CheckMXRecord

Description:	Makes the application check the MX record of email addresses before sending an email or submitting a telephone or email ticket.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CheckMXRecord'} =  '1';`

2.1.16. CheckMXRecord::Nameserver

Description:	Defines the address of a dedicated DNS server, if necessary, for the "CheckMXRecord" look-ups.
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	`$Self->{'CheckMXRecord::Nameserver'} =  'ns.example.com';`

2.1.17. CheckEmailAddresses

Description:	Makes the application check the syntax of email addresses.

Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CheckEmailAddresses'} =  '1';`

2.1.18. CheckEmailValidAddress

Description:	Defines a regular expression that excludes some addresses from the syntax check (if "CheckEmailAddresses" is set to "Yes"). Please enter a regex in this field for email addresses, that aren't syntactically valid, but are necessary for the system (i.e. "root@localhost").
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CheckEmailValidAddress'} =  '^(root@localhost\|admin@localhost)$';`

2.1.19. CheckEmailInvalidAddress

Description:	Defines a regular expression that filters all email addresses that should not be used in the application.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CheckEmailInvalidAddress'} =  '@(example)\\.(..\|...)$';`

2.1.20. CGILogPrefix

Description:	Specifies the text that should appear in the log file to denote a CGI script entry.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CGILogPrefix'} =  'OTRS-CGI';`

2.1.21. DemoSystem

Description:	Runs the system in "Demo" mode. If set to "Yes", agents can change preferences, such as selection of language and theme via the agent web interface. These changes are only valid for the current session. It will not be possible for agents to change their passwords.

Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'DemoSystem'} =  '0';`

2.1.22. SwitchToUser

Description:	Allows the administrators to login as other users, via the users administration panel.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SwitchToUser'} =  '0';`

2.1.23. SwitchToCustomer

Description:	Allows the administrators to login as other customers, via the customer user administration panel.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SwitchToCustomer'} =  '0';`

2.1.24. SwitchToCustomer::PermissionGroup

Description:	Specifies the group where the user needs rw permissions so that he can access the "SwitchToCustomer" feature.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SwitchToCustomer::PermissionGroup'} =  'admin';`

2.1.25. NotificationSenderName

Description:	Specifies the name that should be used by the application when sending notifications. The sender name is used to build the complete display name for the notification master (i.e. "OTRS Notification Master" otrs@your.example.com). Notifications are messages such as en::Customer::QueueUpdate or en::Agent::Move.
Group:	Framework

SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'NotificationSenderName'} =  'OTRS Notification Master';`

2.1.26. NotificationSenderEmail

Description:	Specifies the email address that should be used by the application when sending notifications. The email address is used to build the complete display name for the notification master (i.e. "OTRS Notification Master" otrs@your.example.com). You can use the OTRS_CONFIG_FQDN variable as set in your configuation, or choose another email address. Notifications are messages such as en::Customer::QueueUpdate or en::Agent::Move.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'NotificationSenderEmail'} =  'otrs@<OTRS_CONFIG_FQDN>';`

2.1.27. System::Customer::Permission

Description:	Defines the standard permissions available for customers within the application. If more permissions are needed, you can enter them here. Permissions must be hard coded to be effective. Please ensure, when adding any of the afore mentioned permissions, that the "rw" permission remains the last entry.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1

Config-Setting:	`$Self->{'System::Customer::Permission'} =  [`
	`  'ro',`
	`  'rw'`
	`];`

2.1.28. LanguageDebug

Description:	Debugs the translation set. If this is set to "Yes" all strings (text) without translations are written to STDERR. This can be helpful when you are creating a new translation file. Otherwise, this option should remain set to "No".
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	`$Self->{'LanguageDebug'} =  '0';`

2.1.29. Secure::DisableBanner

Description:	If enabled, the OTRS version tag will be removed from the Webinterface, the HTTP headers and the X-Headers of outgoing mails.
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Secure::DisableBanner'} =  '0';`

2.1.30. StandardTemplate2QueueByCreating

Description:	List of default Standard Templates which are assigned automatically to new Queues upon creation.
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0

Config-Setting:	`$Self->{'StandardTemplate2QueueByCreating'} =  [`
	`  ''`
	`];`

2.2. Core::Cache

2.2.1. Cache::Module

Description:	Selects the cache backend to use.
Group:	Framework
SubGroup:	Core::Cache
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Cache::Module'} =  'Kernel::System::Cache::FileStorable';`

2.2.2. Cache::InMemory

Description:	Should the cache data be help in memory?
Group:	Framework
SubGroup:	Core::Cache
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Cache::InMemory'} =  '1';`

2.2.3. Cache::InBackend

Description:	Should the cache data be stored in the selected cache backend?

Group:	Framework
SubGroup:	Core::Cache
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Cache::InBackend'} =  '1';`

2.2.4. Cache::SubdirLevels

Description:	Specify how many sub directory levels to use when creating cache files. This should prevent too many cache files being in one directory.
Group:	Framework
SubGroup:	Core::Cache
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Cache::SubdirLevels'} =  '2';`

2.3. Core::CustomerCompany

2.3.1. CustomerCompany::EventModulePost###100-UpdateCustomerUsers

Description:	Event module that updates customer users after an update of the Customer.
Group:	Framework
SubGroup:	Core::CustomerCompany
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerCompany::EventModulePost'}->{'100-UpdateCustomerUsers'} =  {`
	`  'Event' => 'CustomerCompanyUpdate',`
	`  'Module' => 'Kernel::System::CustomerCompany::Event::CustomerUserUpdate',`
	`  'Transaction' => '0'`
	`};`

2.4. Core::CustomerUser

2.4.1. CustomerUser::EventModulePost###100-UpdateServiceMembership

Description:	Event module that updates customer user service membership if login changes.
Group:	Framework
SubGroup:	Core::CustomerUser
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerUser::EventModulePost'}->{'100-UpdateServiceMembership'} =  {`
	`  'Event' => 'CustomerUserUpdate',`
	`  'Module' => 'Kernel::System::CustomerUser::Event::ServiceMemberUpdate',`

	`  'Transaction' => '0'`
	`};`

2.5. Core::LinkObject

2.5.1. LinkObject::ViewMode

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

2.5.2. LinkObject::Type###Normal

Description:	Defines the link type 'Normal'. If the source name and the target name contain the same value, the resulting link is a non-directional one; otherwise, the result is a directional link.
Group:	Framework
SubGroup:	Core::LinkObject
Valid:	1
Required:	1

Config-Setting:	`$Self->{'LinkObject::Type'}->{'Normal'} =  {`
	`  'SourceName' => 'Normal',`
	`  'TargetName' => 'Normal'`
	`};`

2.5.3. LinkObject::Type###ParentChild

Description:	Defines the link type 'ParentChild'. If the source name and the target name contain the same value, the resulting link is a non-directional one; otherwise, the result is a directional link.
Group:	Framework
SubGroup:	Core::LinkObject
Valid:	1
Required:	1

Config-Setting:	`$Self->{'LinkObject::Type'}->{'ParentChild'} =  {`
	`  'SourceName' => 'Parent',`
	`  'TargetName' => 'Child'`
	`};`

2.5.4. LinkObject::TypeGroup###0001

Description:	Defines the link type groups. The link types of the same group cancel one another. Example: If ticket A is linked per a 'Normal' link with ticket B, then these tickets could not be additionally linked with link of a 'ParentChild' relationship.

Group:	Framework
SubGroup:	Core::LinkObject
Valid:	1
Required:	0

Config-Setting:	`$Self->{'LinkObject::TypeGroup'}->{'0001'} =  [`
	`  'Normal',`
	`  'ParentChild'`
	`];`

2.6. Core::Log

2.6.1. LogModule

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

2.6.2. LogModule::SysLog::Facility

Description:	If "SysLog" was selected for LogModule, a special log facility can be specified.
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	`$Self->{'LogModule::SysLog::Facility'} =  'user';`

2.6.3. LogModule::SysLog::LogSock

Description:	If "SysLog" was selected for LogModule, a special log sock can be specified (on solaris you may need to use 'stream').
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	`$Self->{'LogModule::SysLog::LogSock'} =  'unix';`

2.6.4. LogModule::SysLog::Charset

Description:	If "SysLog" was selected for LogModule, the charset that should be used for logging can be specified.

Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	`$Self->{'LogModule::SysLog::Charset'} =  'utf-8';`

2.6.5. LogModule::LogFile

Description:	If "file" was selected for LogModule, a logfile must be specified. If the file doesn't exist, it will be created by the system.
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	`$Self->{'LogModule::LogFile'} =  '/tmp/otrs.log';`

2.6.6. LogModule::LogFile::Date

Description:	Adds a suffix with the actual year and month to the OTRS log file. A logfile for every month will be created.
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	`$Self->{'LogModule::LogFile::Date'} =  '0';`

2.7. Core::MIME-Viewer

2.7.1. MIME-Viewer###application/excel

Description:	Specifies the path to the converter that allows the view of Microsoft Excel files, in the web interface.
Group:	Framework
SubGroup:	Core::MIME-Viewer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'MIME-Viewer'}->{'application/excel'} =  'xlhtml';`

2.7.2. MIME-Viewer###application/msword

Description:	Specifies the path to the converter that allows the view of Microsoft Word files, in the web interface.
Group:	Framework
SubGroup:	Core::MIME-Viewer

Valid:	0
Required:	0
Config-Setting:	`$Self->{'MIME-Viewer'}->{'application/msword'} = 'wvWare';`

2.7.3. MIME-Viewer###application/pdf

Description:	Specifies the path to the converter that allows the view of PDF documents, in the web interface.
Group:	Framework
SubGroup:	Core::MIME-Viewer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'MIME-Viewer'}->{'application/pdf'} = 'pdftohtml -stdout -i';`

2.7.4. MIME-Viewer###text/xml

Description:	Specifies the path to the converter that allows the view of XML files, in the web interface.
Group:	Framework
SubGroup:	Core::MIME-Viewer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'MIME-Viewer'}->{'text/xml'} = '<OTRS_CONFIG_Home>/scripts/tools/xml2html.pl';`

2.8. Core::MirrorDB

2.8.1. Core::MirrorDB::DSN

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:	`$Self->{'Core::MirrorDB::DSN'} = 'DBI:mysql:database=mirrordb;host=mirrordbhost';`

2.8.2. Core::MirrorDB::User

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

Valid:	0
Required:	0
Config-Setting:	`$Self->{'Core::MirrorDB::User'} =  'some_user';`

2.8.3. Core::MirrorDB::Password

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

2.9. Core::PDF

2.9.1. PDF

Description:	Enables PDF output. The CPAN module PDF::API2 is required, if not installed, PDF output will be disabled.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PDF'} =  '1';`

2.9.2. PDF::LogoFile

Description:	Specifies the path of the file for the logo in the page header (gif|jpg|png, 700 x 100 pixel).
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PDF::LogoFile'} =  '<OTRS_CONFIG_Home>/var/logo-otrs.png';`

2.9.3. PDF::PageSize

Description:	Defines the standard size of PDF pages.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PDF::PageSize'} =  'a4';`

2.9.4. PDF::MaxPages

Description:	Defines the maximum number of pages per PDF file.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PDF::MaxPages'} =  '100';`

2.9.5. PDF::TTFontFile###Proportional

Description:	Defines the path and TTF-File to handle proportional font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PDF::TTFontFile'}->{'Proportional'} =  'DejaVuSans.ttf';`

2.9.6. PDF::TTFontFile###ProportionalBold

Description:	Defines the path and TTF-File to handle bold proportional font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PDF::TTFontFile'}->{'ProportionalBold'} =  'DejaVuSans-Bold.ttf';`

2.9.7. PDF::TTFontFile###ProportionalItalic

Description:	Defines the path and TTF-File to handle italic proportional font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	0
Required:	0
Config-Setting:	`$Self->{'PDF::TTFontFile'}->{'ProportionalItalic'} =  'DejaVuSans-Oblique.ttf';`

2.9.8. PDF::TTFontFile###ProportionalBoldItalic

Description:	Defines the path and TTF-File to handle bold italic proportional font in PDF documents.

Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PDF::TTFontFile'}->{'ProportionalBoldItalic'} = 'DejaVuSans-BoldOblique.ttf';`

2.9.9. PDF::TTFontFile###Monospaced

Description:	Defines the path and TTF-File to handle monospaced font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PDF::TTFontFile'}->{'Monospaced'} =  'DejaVuSansMono.ttf';`

2.9.10. PDF::TTFontFile###MonospacedBold

Description:	Defines the path and TTF-File to handle bold monospaced font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	0
Required:	0
Config-Setting:	`$Self->{'PDF::TTFontFile'}->{'MonospacedBold'} =  'DejaVuSansMono-Bold.ttf';`

2.9.11. PDF::TTFontFile###MonospacedItalic

Description:	Defines the path and TTF-File to handle italic monospaced font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	0
Required:	0
Config-Setting:	`$Self->{'PDF::TTFontFile'}->{'MonospacedItalic'} =  'DejaVuSansMono-Oblique.ttf';`

2.9.12. PDF::TTFontFile###MonospacedBoldItalic

Description:	Defines the path and TTF-File to handle bold italic monospaced font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF

Valid:	0
Required:	0

Config-Setting:	`$Self->{'PDF::TTFontFile'}->{'MonospacedBoldItalic'} =`
	`'DejaVuSansMono-BoldOblique.ttf';`

2.10. Core::Package

2.10.1. Package::FileUpload

Description:	Enables file upload in the package manager frontend.
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Package::FileUpload'} =  '1';`

2.10.2. Package::RepositoryRoot

Description:	Defines the location to get online repository list for additional packages. The first available result will be used.
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Package::RepositoryRoot'} =  [`
	`  'http://ftp.otrs.org/pub/otrs/misc/packages/repository.xml'`
	`];`

2.10.3. Package::RepositoryList

Description:	Defines the list of online repositories. Another installations can be used as repository, for example: Key="http://example.com/otrs/public.pl?Action=PublicRepository;File=" and Content="Some Name".
Group:	Framework
SubGroup:	Core::Package
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Package::RepositoryList'} =  {`
	`  'ftp://ftp.example.com/pub/otrs/misc/packages/' => '[Example] ftp://`
	`ftp.example.com/'`
	`};`

2.10.4. Package::RepositoryAccessRegExp

Description:	Defines the IP regular expression for accessing the local repository. You need to enable this to have access to your local repository and the package::RepositoryList is required on the remote host.

Group:	Framework
SubGroup:	Core::Package
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Package::RepositoryAccessRegExp'} =  '127\\.0\\.0\\.1';`

2.10.5. Package::Timeout

Description:	Sets the timeout (in seconds) for package downloads. Overwrites "WebUserAgent::Timeout".
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Package::Timeout'} =  '120';`

2.10.6. Package::Proxy

Description:	Fetches packages via proxy. Overwrites "WebUserAgent::Proxy".
Group:	Framework
SubGroup:	Core::Package
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Package::Proxy'} =  'http://proxy.sn.no:8001/';`

2.10.7. Package::ShowFeatureAddons

Description:	Toggles display of OTRS FeatureAddons list in PackageManager.
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Package::ShowFeatureAddons'} =  '1';`

2.10.8. Package::EventModulePost###99-SupportDataSend

Description:	Package event module file a scheduler task for update registration.
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Package::EventModulePost'}->{'99-SupportDataSend'} =  {`
	`  'Event' => '(PackageInstall|PackageReinstall|PackageUpgrade|`
	`PackageUninstall)',`

	`'Module' => 'Kernel::System::Package::Event::SupportDataSend',`
	`'Transaction' => '1'`
	`};`

2.11. Core::PerformanceLog

2.11.1. PerformanceLog

Description:	Enables performance log (to log the page response time). It will affect the system performance. Frontend::Module###AdminPerformanceLog must be enabled.
Group:	Framework
SubGroup:	Core::PerformanceLog
Valid:	1
Required:	0
Config-Setting:	`$Self->{'PerformanceLog'} =  '0';`

2.11.2. PerformanceLog::File

Description:	Specifies the path of the file for the performance log.
Group:	Framework
SubGroup:	Core::PerformanceLog
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PerformanceLog::File'} =  '<OTRS_CONFIG_Home>/var/log/Performance.log';`

2.11.3. PerformanceLog::FileMax

Description:	Defines the maximum size (in MB) of the log file.
Group:	Framework
SubGroup:	Core::PerformanceLog
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PerformanceLog::FileMax'} =  '25';`

2.12. Core::ReferenceData

2.12.1. ReferenceData::OwnCountryList

Description:	This setting allows you to override the built-in country list with your own list of countries. This is particularly handy if you just want to use a small select group of countries.
Group:	Framework
SubGroup:	Core::ReferenceData
Valid:	0
Required:	0

Config-Setting:	`$Self->{'ReferenceData::OwnCountryList'} =  {`
	`  'AT' => 'Austria',`
	`  'CH' => 'Switzerland',`
	`  'DE' => 'Germany'`
	`};`

2.13. Core::SOAP

2.13.1. SOAP::User

Description:	Defines the username to access the SOAP handle (bin/cgi-bin/rpc.pl).
Group:	Framework
SubGroup:	Core::SOAP
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SOAP::User'} =  'some_user';`

2.13.2. SOAP::Password

Description:	Defines the password to access the SOAP handle (bin/cgi-bin/rpc.pl).
Group:	Framework
SubGroup:	Core::SOAP
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SOAP::Password'} =  'some_pass';`

2.13.3. SOAP::Keep-Alive

Description:	Enable keep-alive connection header for SOAP responses.
Group:	Framework
SubGroup:	Core::SOAP
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SOAP::Keep-Alive'} =  '0';`

2.14. Core::Sendmail

2.14.1. SendmailModule

Description:	Defines the module to send emails. "Sendmail" directly uses the sendmail binary of your operating system. Any of the "SMTP" mechanisms use a specified (external) mailserver. "DoNotSendEmail" doesn't send emails and it is useful for test systems.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	1

Required:	1
Config-Setting:	`$Self->{'SendmailModule'} =  'Kernel::System::Email::Sendmail';`

2.14.2. SendmailModule::CMD

Description:	If "Sendmail" was selected as SendmailModule, the location of the sendmail binary and the needed options must be specified.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SendmailModule::CMD'} =  '/usr/sbin/sendmail -i -f';`

2.14.3. SendmailModule::Host

Description:	If any of the "SMTP" mechanisms was selected as SendmailModule, the mailhost that sends out the mails must be specified.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SendmailModule::Host'} =  'mail.example.com';`

2.14.4. SendmailModule::Port

Description:	If any of the "SMTP" mechanisms was selected as SendmailModule, the port where your mailserver is listening for incoming connections must be specified.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SendmailModule::Port'} =  '25';`

2.14.5. SendmailModule::AuthUser

Description:	If any of the "SMTP" mechanisms was selected as SendmailModule, and authentication to the mail server is needed, an username must be specified.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SendmailModule::AuthUser'} =  'MailserverLogin';`

2.14.6. SendmailModule::AuthPassword

Description:	If any of the "SMTP" mechanisms was selected as SendmailModule, and authentication to the mail server is needed, a password must be specified.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SendmailModule::AuthPassword'} =  'MailserverPassword';`

2.14.7. SendmailBcc

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

2.14.8. SendmailEnvelopeFrom

Description:	If set, this address is used as envelope sender in outgoing messages (not notifications - see below). If no address is specified, the envelope sender is equal to queue e-mail address.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SendmailEnvelopeFrom'} =  '';`

2.14.9. SendmailNotificationEnvelopeFrom

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

2.14.10. SendmailEncodingForce

Description:	Forces encoding of outgoing emails (7bit|8bit|quoted-printable|base64).

Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SendmailEncodingForce'} =  'base64';`

2.15. Core::Session

2.15.1. SessionModule

Description:	Defines the module used to store the session data. With "DB" the frontend server can be splitted from the db server. "FS" is faster.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionModule'} =  'Kernel::System::AuthSession::DB';`

2.15.2. SessionName

Description:	Defines the name of the session key. E.g. Session, SessionID or OTRS.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionName'} =  'OTRSAgentInterface';`

2.15.3. CustomerPanelSessionName

Description:	Defines the name of the key for customer sessions.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelSessionName'} =  'OTRSCustomerInterface';`

2.15.4. SessionCheckRemoteIP

Description:	Turns on the remote ip address check. It should be set to "No" if the application is used, for example, via a proxy farm or a dialup connection, because the remote ip address is mostly different for the requests.
Group:	Framework
SubGroup:	Core::Session
Valid:	1

Required:	1
Config-Setting:	`$Self->{'SessionCheckRemoteIP'} =  '1';`

2.15.5. SessionDeleteIfNotRemoteID

Description:	Deletes a session if the session id is used with an invalid remote IP address.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionDeleteIfNotRemoteID'} =  '1';`

2.15.6. SessionMaxTime

Description:	Defines the maximal valid time (in seconds) for a session id.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionMaxTime'} =  '57600';`

2.15.7. SessionMaxIdleTime

Description:	Sets the inactivity time (in seconds) to pass before a session is killed and a user is loged out.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionMaxIdleTime'} =  '21600';`

2.15.8. SessionActiveTime

Description:	Sets the time (in seconds) a user is marked as active.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionActiveTime'} =  '600';`

2.15.9. SessionDeleteIfTimeToOld

Description:	Deletes requested sessions if they have timed out.
Group:	Framework

SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionDeleteIfTimeToOld'} =  '1';`

2.15.10. SessionUseCookie

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

2.15.11. SessionUseCookieAfterBrowserClose

Description:	Stores cookies after the browser has been closed.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionUseCookieAfterBrowserClose'} =  '0';`

2.15.12. SessionCSRFProtection

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:	`$Self->{'SessionCSRFProtection'} =  '1';`

2.15.13. AgentSessionLimit

Description:	Sets the maximum number of active agents within the timespan defined in SessionActiveTime.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	0

Config-Setting:	`$Self->{'AgentSessionLimit'} =  '100';`

2.15.14. AgentSessionPerUserLimit

Description:	Sets the maximum number of active sessions per agent within the timespan defined in SessionActiveTime.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	0
Config-Setting:	`$Self->{'AgentSessionPerUserLimit'} =  '20';`

2.15.15. CustomerSessionLimit

Description:	Sets the maximum number of active customers within the timespan defined in SessionActiveTime.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerSessionLimit'} =  '100';`

2.15.16. CustomerSessionPerUserLimit

Description:	Sets the maximum number of active sessions per customers within the timespan defined in SessionActiveTime.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerSessionPerUserLimit'} =  '20';`

2.15.17. SessionDir

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

2.15.18. SessionTable

Description:	If "DB" was selected for SessionModule, a table in database where session data will be stored must be specified.

Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionTable'} =  'sessions';`

2.16. Core::SpellChecker

2.16.1. SpellChecker

Description:	Enables spell checker support.
Group:	Framework
SubGroup:	Core::SpellChecker
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SpellChecker'} =  '0';`

2.16.2. SpellCheckerBin

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

2.16.3. SpellCheckerDictDefault

Description:	Defines the default spell checker dictionary.
Group:	Framework
SubGroup:	Core::SpellChecker
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SpellCheckerDictDefault'} =  'english';`

2.16.4. SpellCheckerIgnore

Description:	Defines a default list of words, that are ignored by the spell checker.
Group:	Framework
SubGroup:	Core::SpellChecker
Valid:	1
Required:	1

Config-Setting:	$Self->{'SpellCheckerIgnore'} = [
	  'www',
	  'webmail',
	  'https',
	  'http',
	  'html',
	  'rfc'
	];

2.17. Core::Stats

2.17.1. Stats::StatsHook

Description:	Sets the stats hook.
Group:	Framework
SubGroup:	Core::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::StatsHook'} =  'Stat#';`

2.17.2. Stats::StatsStartNumber

Description:	Start number for statistics counting. Every new stat increments this number.
Group:	Framework
SubGroup:	Core::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::StatsStartNumber'} =  '10000';`

2.17.3. Stats::MaxXaxisAttributes

Description:	Defines the default maximum number of X-axis attributes for the time scale.
Group:	Framework
SubGroup:	Core::Stats
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Stats::MaxXaxisAttributes'} =  '1000';`

2.18. Core::Stats::Graph

2.18.1. Stats::Graph::t_margin

Description:	Specifies the top margin of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Stats::Graph::t_margin'} =  '10';`

2.18.2. Stats::Graph::l_margin

Description:	Specifies the left margin of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::l_margin'} =  '10';`

2.18.3. Stats::Graph::b_margin

Description:	Specifies the bottom margin of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::b_margin'} =  '10';`

2.18.4. Stats::Graph::r_margin

Description:	Specifies the right margin of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::r_margin'} =  '20';`

2.18.5. Stats::Graph::bgclr

Description:	Specifies the background color of the picture.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::bgclr'} =  'white';`

2.18.6. Stats::Graph::transparent

Description:	Makes the picture transparent.
Group:	Framework

SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::transparent'} =  '0';`

2.18.7. Stats::Graph::fgclr

Description:	Specifies the border color of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::fgclr'} =  'black';`

2.18.8. Stats::Graph::boxclr

Description:	Specifies the background color of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::boxclr'} =  'white';`

2.18.9. Stats::Graph::accentclr

Description:	Specifies the border color of the legend.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::accentclr'} =  'black';`

2.18.10. Stats::Graph::legendclr

Description:	Specifies the text color of the legend.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::legendclr'} =  'black';`

2.18.11. Stats::Graph::textclr

Description:	Specifies the text color of the chart (e. g. caption).

Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::textclr'} =  'black';`

2.18.12. Stats::Graph::dclrs

Description:	Defines the colors for the graphs.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Stats::Graph::dclrs'} =  [`
	`  'red',`
	`  'green',`
	`  'blue',`
	`  'yellow',`
	`  'purple',`
	`  'orange',`
	`  'pink',`
	`  'marine',`
	`  'cyan',`
	`  'lgray',`
	`  'lblue',`
	`  'lyellow',`
	`  'lgreen',`
	`  'lred',`
	`  'lpurple',`
	`  'lorange',`
	`  'lbrown'`
	`];`

2.18.13. Stats::Graph::TitleFont

Description:	Defines the title font in graphs (place custom fonts in var/fonts).
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::TitleFont'} =  'DejaVuSans-Bold.ttf';`

2.18.14. Stats::Graph::LegendFont

Description:	Defines the legend font in graphs (place custom fonts in var/fonts).
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Stats::Graph::LegendFont'} =  'DejaVuSans.ttf';`

2.18.15. Stats::Graph::line_width

Description:	Defines the boldness of the line drawed by the graph.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::line_width'} =  '1';`

2.18.16. Stats::Graph::legend_placement

Description:	Defines the placement of the legend. This should be a two letter key of the form: 'B[LCR]|R[TCB]'. The first letter indicates the placement (Bottom or Right), and the second letter the alignment (Left, Right, Center, Top, or Bottom).
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::legend_placement'} =  'BC';`

2.18.17. Stats::Graph::legend_spacing

Description:	Defines the spacing of the legends.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::legend_spacing'} =  '4';`

2.18.18. Stats::Graph::legend_marker_width

Description:	Defines the width of the legend.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::legend_marker_width'} =  '12';`

2.18.19. Stats::Graph::legend_marker_height

Description:	Defines the height of the legend.

Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::legend_marker_height'} =  '8';`

2.19. Core::Time

2.19.1. TimeInputFormat

Description:	Defines the date input format used in forms (option or input fields).
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeInputFormat'} =  'Option';`

2.19.2. TimeShowAlwaysLong

Description:	Shows time in long format (days, hours, minutes), if set to "Yes"; or in short format (days, hours), if set to "No".
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeShowAlwaysLong'} =  '0';`

2.19.3. TimeZone

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

2.19.4. TimeZoneUser

Description:	Sets the user time zone per user (required a system with UTC as system time and UTC under TimeZone). Otherwise this is a diff time to the local time.
Group:	Framework
SubGroup:	Core::Time
Valid:	1

Required:	0
Config-Setting:	`$Self->{'TimeZoneUser'} =  '0';`

2.19.5. TimeZoneUserBrowserAutoOffset

Description:	Sets the user time zone per user based on java script / browser time zone offset feature at login time.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	0
Config-Setting:	`$Self->{'TimeZoneUserBrowserAutoOffset'} =  '1';`

2.19.6. CalendarWeekDayStart

Description:	Define the start day of the week for the date picker.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CalendarWeekDayStart'} =  '1';`

2.19.7. TimeVacationDays

Description:	Adds the permanent vacation days. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1

Config-Setting:	`$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'`
	`  }`
	`};`

2.19.8. TimeVacationDaysOneTime

Description:	Adds the one time vacation days. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).

<table>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Core::Time</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>1</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'TimeVacationDaysOneTime'} = {
  '2004' => {
    '1' => {
      '1' => 'test'
    }
  }
};
</pre></td></tr>
</table>

2.19.9. TimeWorkingHours

<table>
<tr><td>Description:</td><td>Defines the hours and week days to count the working time.</td></tr>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Core::Time</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>1</td></tr>
<tr><td>Config-Setting:</td><td><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',
</pre></td></tr>
</table>

```
    '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'
  ]
};
```

2.19.10. TimeShowCompleteDescription

Description:	Shows time use complete description (days, hours, minutes), if set to "Yes"; or just first letter (d, h, m), if set to "No".
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeShowCompleteDescription'} =  '0';`

2.20. Core::Time::Calendar1

2.20.1. TimeZone::Calendar1Name

Description:	Defines the name of the indicated calendar.
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1

Config-Setting:	`$Self->{'TimeZone::Calendar1Name'} =  'Calendar Name 1';`

2.20.2. TimeZone::Calendar1

Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar1'} =  '0';`

2.20.3. TimeVacationDays::Calendar1

Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1

Config-Setting:	`$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'`
	`  }`
	`};`

2.20.4. TimeVacationDaysOneTime::Calendar1

Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1

Config-Setting:	`$Self->{'TimeVacationDaysOneTime::Calendar1'} =  {`
	`  '2004' => {`
	`    '1' => {`
	`      '1' => 'test'`
	`    }`
	`  }`
	`};`

2.20.5. TimeWorkingHours::Calendar1

Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1

Config-Setting:

```
$Self->{'TimeWorkingHours::Calendar1'} =  {
  'Fri' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Mon' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Sat' => [],
  'Sun' => [],
  'Thu' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Tue' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Wed' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ]
};
```

2.21. Core::Time::Calendar2

2.21.1. TimeZone::Calendar2Name

Description:	Defines the name of the indicated calendar.
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar2Name'} =  'Calendar Name 2';`

2.21.2. TimeZone::Calendar2

Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar2'} =  '0';`

2.21.3. TimeVacationDays::Calendar2

Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1
Config-Setting:	`$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'`
	`  }`
	`};`

2.21.4. TimeVacationDaysOneTime::Calendar2

Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1

Config-Setting:	` $Self->{'TimeVacationDaysOneTime::Calendar2'} =  {`
	`  '2004' => {`
	`    '1' => {`
	`      '1' => 'test'`
	`    }`
	`  }`
	`};`

2.21.5. TimeWorkingHours::Calendar2

Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1

Config-Setting:	` $Self->{'TimeWorkingHours::Calendar2'} =  {`
	`  'Fri' => [`
	`    '8',`
	`    '9',`
	`    '10',`
	`    '11',`
	`    '12',`
	`    '13',`
	`    '14',`
	`    '15',`
	`    '16',`
	`    '17',`
	`    '18',`
	`    '19',`
	`    '20'`
	`  ],`
	`  'Mon' => [`
	`    '8',`
	`    '9',`
	`    '10',`

```
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Sat' => [],
  'Sun' => [],
  'Thu' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Tue' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Wed' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ]
};
```

2.22. Core::Time::Calendar3

2.22.1. TimeZone::Calendar3Name

Description:	Defines the name of the indicated calendar.
Group:	Framework

SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar3Name'} =  'Calendar Name 3';`

2.22.2. TimeZone::Calendar3

Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar3'} =  '0';`

2.22.3. TimeVacationDays::Calendar3

Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1

Config-Setting:	`$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'`
	`  }`
	`};`

2.22.4. TimeVacationDaysOneTime::Calendar3

Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1

Config-Setting:	`$Self->{'TimeVacationDaysOneTime::Calendar3'} =  {`
	`  '2004' => {`
	`    '1' => {`
	`      '1' => 'test'`
	`    }`
	`  }`

```
};
```

2.22.5. TimeWorkingHours::Calendar3

Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1

Config-Setting:

```
$Self->{'TimeWorkingHours::Calendar3'} =  {
 'Fri' => [
   '8',
   '9',
   '10',
   '11',
   '12',
   '13',
   '14',
   '15',
   '16',
   '17',
   '18',
   '19',
   '20'
 ],
 'Mon' => [
   '8',
   '9',
   '10',
   '11',
   '12',
   '13',
   '14',
   '15',
   '16',
   '17',
   '18',
   '19',
   '20'
 ],
 'Sat' => [],
 'Sun' => [],
 'Thu' => [
   '8',
   '9',
   '10',
   '11',
   '12',
   '13',
   '14',
   '15',
   '16',
   '17',
   '18',
   '19',
   '20'
 ],
 'Tue' => [
   '8',
   '9',
   '10',
   '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Wed' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ]
};
```

2.23. Core::Time::Calendar4

2.23.1. TimeZone::Calendar4Name

Description:	Defines the name of the indicated calendar.
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar4Name'} =  'Calendar Name 4';`

2.23.2. TimeZone::Calendar4

Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar4'} =  '0';`

2.23.3. TimeVacationDays::Calendar4

Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1

Required:	1

Config-Setting:

```
$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'
  }
};
```

2.23.4. TimeVacationDaysOneTime::Calendar4

Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1

Config-Setting:

```
$Self->{'TimeVacationDaysOneTime::Calendar4'} =  {
  '2004' => {
    '1' => {
      '1' => 'test'
    }
  }
};
```

2.23.5. TimeWorkingHours::Calendar4

Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1

Config-Setting:

```
$Self->{'TimeWorkingHours::Calendar4'} =  {
  'Fri' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Mon' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Sat' => [],
  'Sun' => [],
  'Thu' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Tue' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Wed' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ]
};
```

2.24. Core::Time::Calendar5

2.24.1. TimeZone::Calendar5Name

Description:	Defines the name of the indicated calendar.

Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar5Name'} =  'Calendar Name 5';`

2.24.2. TimeZone::Calendar5

Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar5'} =  '0';`

2.24.3. TimeVacationDays::Calendar5

<table>
<tr><td>Description:</td><td>Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).</td></tr>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Core::Time::Calendar5</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>1</td></tr>
<tr><td>Config-Setting:</td><td><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></td></tr>
</table>

2.24.4. TimeVacationDaysOneTime::Calendar5

<table>
<tr><td>Description:</td><td>Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).</td></tr>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Core::Time::Calendar5</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>1</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'TimeVacationDaysOneTime::Calendar5'} = {
  '2004' => {
    '1' => {
      '1' => 'test'
    }
</pre></td></tr>
</table>

```
  }
};
```

2.24.5. TimeWorkingHours::Calendar5

Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1

Config-Setting:

```
$Self->{'TimeWorkingHours::Calendar5'} =  {
  'Fri' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Mon' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Sat' => [],
  'Sun' => [],
  'Thu' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Tue' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Wed' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ]
};
```

2.25. Core::Time::Calendar6

2.25.1. TimeZone::Calendar6Name

Description:	Defines the name of the indicated calendar.
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar6Name'} =  'Calendar Name 6';`

2.25.2. TimeZone::Calendar6

Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar6'} =  '0';`

2.25.3. TimeVacationDays::Calendar6

Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1

Required:	1

Config-Setting:

```
$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'
  }
};
```

2.25.4. TimeVacationDaysOneTime::Calendar6

Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1

Config-Setting:

```
$Self->{'TimeVacationDaysOneTime::Calendar6'} =  {
  '2004' => {
    '1' => {
      '1' => 'test'
    }
  }
};
```

2.25.5. TimeWorkingHours::Calendar6

Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1

Config-Setting:

```
$Self->{'TimeWorkingHours::Calendar6'} =  {
  'Fri' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Mon' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Sat' => [],
  'Sun' => [],
  'Thu' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Tue' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Wed' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ]
};
```

2.26. Core::Time::Calendar7

2.26.1. TimeZone::Calendar7Name

Description:	Defines the name of the indicated calendar.

Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar7Name'} =  'Calendar Name 7';`

2.26.2. TimeZone::Calendar7

Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar7'} =  '0';`

2.26.3. TimeVacationDays::Calendar7

Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1

Config-Setting:	`$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'`
	`  }`
	`};`

2.26.4. TimeVacationDaysOneTime::Calendar7

Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1

Config-Setting:	`$Self->{'TimeVacationDaysOneTime::Calendar7'} =  {`
	`  '2004' => {`
	`    '1' => {`
	`      '1' => 'test'`
	`    }`

```
  }
};
```

2.26.5. TimeWorkingHours::Calendar7

Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'TimeWorkingHours::Calendar7'} =  {
  'Fri' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Mon' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Sat' => [],
  'Sun' => [],
  'Thu' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Tue' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Wed' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ]
};
```

2.27. Core::Time::Calendar8

2.27.1. TimeZone::Calendar8Name

Description:	Defines the name of the indicated calendar.
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar8Name'} =  'Calendar Name 8';`

2.27.2. TimeZone::Calendar8

Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar8'} =  '0';`

2.27.3. TimeVacationDays::Calendar8

Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1

<table>
<tr><td>Required:</td><td>1</td></tr>
<tr><td>Config-Setting:</td><td><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></td></tr>
</table>

2.27.4. TimeVacationDaysOneTime::Calendar8

<table>
<tr><td>Description:</td><td>Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).</td></tr>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Core::Time::Calendar8</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>1</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'TimeVacationDaysOneTime::Calendar8'} = {
  '2004' => {
    '1' => {
      '1' => 'test'
    }
  }
};
</pre></td></tr>
</table>

2.27.5. TimeWorkingHours::Calendar8

<table>
<tr><td>Description:</td><td>Defines the hours and week days of the indicated calendar, to count the working time.</td></tr>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Core::Time::Calendar8</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>1</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'TimeWorkingHours::Calendar8'} = {
  'Fri' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Mon' => [
    '8',
</pre></td></tr>
</table>

```
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Sat' => [],
  'Sun' => [],
  'Thu' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Tue' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Wed' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ]
};
```

2.28. Core::Time::Calendar9

2.28.1. TimeZone::Calendar9Name

Description:	Defines the name of the indicated calendar.

Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar9Name'} =  'Calendar Name 9';`

2.28.2. TimeZone::Calendar9

Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar9'} =  '0';`

2.28.3. TimeVacationDays::Calendar9

Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1

Config-Setting:	`$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'`
	`  }`
	`};`

2.28.4. TimeVacationDaysOneTime::Calendar9

Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeVacationDaysOneTime::Calendar9'} =  {`

<table>
<tr><td></td><td><pre> '2004' => {
    '1' => {
      '1' => 'test'
    }
  }
};</pre></td></tr>
</table>

2.28.5. TimeWorkingHours::Calendar9

<table>
<tr><td>Description:</td><td>Defines the hours and week days of the indicated calendar, to count the working time.</td></tr>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Core::Time::Calendar9</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>1</td></tr>
<tr><td>Config-Setting:</td><td><pre>$Self->{'TimeWorkingHours::Calendar9'} = {
  'Fri' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Mon' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Sat' => [],
  'Sun' => [],
  'Thu' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],</pre></td></tr>
</table>

```
  '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'
  ]
};
```

2.29. Core::Web

2.29.1. Frontend::WebPath

Description:	Defines the URL base path of icons, CSS and Java Script.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::WebPath'} =  '/otrs-web/';`

2.29.2. Frontend::ImagePath

Description:	Defines the URL image path of icons for navigation.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Frontend::ImagePath'} =`
	`'<OTRS_CONFIG_Frontend::WebPath>skins/Agent/default/img/';`

2.29.3. Frontend::CSSPath

Description:	Defines the URL CSS path.

Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Frontend::CSSPath'} =`
	`'<OTRS_CONFIG_Frontend::WebPath>css/';`

2.29.4. Frontend::JavaScriptPath

Description:	Defines the URL java script path.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Frontend::JavaScriptPath'} =`
	`'<OTRS_CONFIG_Frontend::WebPath>js/';`

2.29.5. Frontend::RichText

Description:	Uses richtext for viewing and editing: articles, salutations, signatures, standard templates, auto responses and notifications.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::RichText'} =  '1';`

2.29.6. Frontend::RichTextPath

Description:	Defines the URL rich text editor path.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Frontend::RichTextPath'} =`
	`'<OTRS_CONFIG_Frontend::WebPath>js/thirdparty/ckeditor-4.3/';`

2.29.7. Frontend::RichTextWidth

Description:	Defines the width for the rich text editor component. Enter number (pixels) or percent value (relative).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Frontend::RichTextWidth'} =  '620';`

2.29.8. Frontend::RichTextHeight

Description:	Defines the height for the rich text editor component. Enter number (pixels) or percent value (relative).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::RichTextHeight'} =  '320';`

2.29.9. Frontend::RichText::DefaultCSS

Description:	Defines the default CSS used in rich text editors.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::RichText::DefaultCSS'} =  'font-family:Geneva,Helvetica,Arial,sans-serif; font-size: 12px;';`

2.29.10. Frontend::RichText::EnhancedMode

Description:	Defines if the enhanced mode should be used (enables use of table, replace, subscript, superscript, paste from word, etc.).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::RichText::EnhancedMode'} =  '0';`

2.29.11. DisableMSIFrameSecurityRestricted

Description:	Disable restricted security for IFrames in IE. May be required for SSO to work in IE8.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	0
Config-Setting:	`$Self->{'DisableMSIFrameSecurityRestricted'} =  '0';`

2.29.12. DefaultViewNewLine

Description:	Automated line break in text messages after x number of chars.

Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'DefaultViewNewLine'} =  '90';`

2.29.13. DefaultViewLines

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

2.29.14. Frontend::AnimationEnabled

Description:	Turns on the animations used in the GUI. If you have problems with these animations (e.g. performance issues), you can turn them off here.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::AnimationEnabled'} =  '1';`

2.29.15. AttachmentDownloadType

Description:	Allows choosing between showing the attachments of a ticket in the browser (inline) or just make them downloadable (attachment).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'AttachmentDownloadType'} =  'attachment';`

2.29.16. WebMaxFileUpload

Description:	Defines the maximal size (in bytes) for file uploads via the browser. Warning: Setting this option to a value which is too low could cause many masks in your OTRS instance to stop working (probably any mask which takes input from the user).
Group:	Framework
SubGroup:	Core::Web

Valid:	1
Required:	1
Config-Setting:	`$Self->{'WebMaxFileUpload'} =  '24000000';`

2.29.17. WebUploadCacheModule

Description:	Selects the module to handle uploads via the web interface. "DB" stores all uploads in the database, "FS" uses the file system.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1

Config-Setting:	`$Self->{'WebUploadCacheModule'} =`
	`'Kernel::System::Web::UploadCache::DB';`

2.29.18. Frontend::Output::FilterText###AAAURL

Description:	Defines the filter that processes the text in the articles, in order to highlight URLs.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Output::FilterText'}->{'AAAURL'} =  {`
	`  'Module' => 'Kernel::Output::HTML::OutputFilterTextURL',`
	`  'Templates' => {`
	`    'AgentTicketZoom' => '1'`
	`  }`
	`};`

2.29.19. Frontend::Themes

Description:	Activates the available themes on the system. Value 1 means active, 0 means inactive.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Themes'} =  {`
	`  'Lite' => '0',`
	`  'Standard' => '1'`
	`};`

2.29.20. Frontend::Output::FilterText###OutputFilterTextAutoLink

Description:	Defines a filter to process the text in the articles, in order to highlight predefined keywords.
Group:	Framework

SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Output::FilterText'}->{'OutputFilterTextAutoLink'}
 =  {
  'Module' => 'Kernel::Output::HTML::OutputFilterTextAutoLink',
  'Templates' => {
    'AgentTicketZoom' => '1'
  }
};
```

2.29.21. Frontend::Output::OutputFilterTextAutoLink###CVE

Description:	Defines a filter for html output to add links behind CVE numbers. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possiblity is to insert the link to the image.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Output::OutputFilterTextAutoLink'}->{'CVE'} =  {
  'RegExp' => [
    '(CVE|CAN)\\-(\\d{3,4})\\-(\\d{2,})'
  ],
  'Templates' => {
    'AgentTicketZoom' => '1'
  },
  '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>'
  }
};
```

2.29.22. Frontend::Output::OutputFilterTextAutoLink###Bugtraq

Description:	Defines a filter for html output to add links behind bugtraq numbers. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possiblity is to insert the link to the image.
Group:	Framework

SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Output::OutputFilterTextAutoLink'}->{'Bugtraq'} =
 {
  'RegExp' => [
    'Bugtraq[\\s\\w\\t]*?ID[\\s\\w\\t]*?:[\\s\\w\\t]*?(\\d{2,8})',
    'Bugtraq[\\s\\w\\t]*?ID[\\s\\w\\t]*?(\\d{2,8})',
    'Bugtraq[\\s\\w\\t]*?:[\\s\\w\\t]*?(\\d{2,8})',
    'Bugtraq[\\s\\w\\t]*?(\\d{2,8})',
    'BID[\\s\\w\\t]*?:[\\s\\w\\t]*?(\\d{2,8})',
    'BID[\\s\\w\\t]*?(\\d{2,8})'
  ],
  'Templates' => {
    'AgentTicketZoom' => '1'
  },
  '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>'
  }
};
```

2.29.23. Frontend::Output::OutputFilterTextAutoLink###MSBulletins

Description:	Defines a filter for html output to add links behind MSBulletin numbers. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possiblity is to insert the link to the image.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Output::OutputFilterTextAutoLink'}-
>{'MSBulletins'} =  {
  'RegExp' => [
    'MS[^A-Za-z]{0,5}(\\d\\d).?(\\d{2,4})'
  ],
  'Templates' => {
    'AgentTicketZoom' => '1'
  },
  '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',
    'Target' => '_blank',
    'URL' => 'http://google.com/search?q=MS<MATCH1>-<MATCH2>'
```

	`  }`
	`};`

2.29.24. Frontend::Output::OutputFilterTextAutoLink###Setting1

Description:	Define a filter for html output to add links behind a defined string. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possiblity is to insert the link to the image.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::Output::OutputFilterTextAutoLink'}->{'Setting1'} =`
	`  {`
	`  'RegExp' => [`
	`    'RegExp'`
	`  ],`
	`  'Templates' => {`
	`    'AgentTicketZoom' => '1'`
	`  },`
	`  'URL1' => {`
	`    'Description' => 'Description',`
	`    'Image' => 'right-small.png',`
	`    'Target' => '_blank',`
	`    'URL' => 'URL'`
	`  },`
	`  'URL2' => {`
	`    'Description' => 'Description',`
	`    'Image' => 'Image',`
	`    'Target' => '_blank',`
	`    'URL' => 'URL'`
	`  }`
	`};`

2.29.25. Frontend::Output::OutputFilterTextAutoLink###Setting2

Description:	Defines a filter for html output to add links behind a defined string. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possiblity is to insert the link to the image.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::Output::OutputFilterTextAutoLink'}->{'Setting2'} =`
	`  {`
	`  'RegExp' => [`
	`    'RegExp'`
	`  ],`
	`  'Templates' => {`
	`    'AgentTicketZoom' => '1'`
	`  },`
	`  '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'
  }
};
```

2.29.26. Loader::Enabled::CSS

Description:	If enabled, OTRS will deliver all CSS files in minified form. WARNING: If you turn this off, there will likely be problems in IE 7, because it cannot load more than 32 CSS files.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Loader::Enabled::CSS'} =  '1';`

2.29.27. Loader::Enabled::JS

Description:	If enabled, OTRS will deliver all JavaScript files in minified form.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Loader::Enabled::JS'} =  '1';`

2.29.28. Loader::Agent::CommonCSS###000-Framework

Description:	List of CSS files to always be loaded for the agent interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Loader::Agent::CommonCSS'}->{'000-Framework'} =  [`
	`'Core.Reset.css',`
	`'Core.Default.css',`
	`'Core.Header.css',`
	`'Core.OverviewControl.css',`
	`'Core.OverviewSmall.css',`
	`'Core.OverviewMedium.css',`
	`'Core.OverviewLarge.css',`
	`'Core.Footer.css',`

<table>
<tr><td></td><td><pre>
  'Core.PageLayout.css',
  'Core.Form.css',
  'Core.Table.css',
  'Core.Widget.css',
  'Core.WidgetMenu.css',
  'Core.TicketDetail.css',
  'Core.Tooltip.css',
  'Core.Dialog.css',
  'Core.Print.css',
  'thirdparty/fontawesome/font-awesome.css'
];
</pre></td></tr>
</table>

2.29.29. Loader::Agent::CommonCSS::IE8###000-Framework

<table>
<tr><td>Description:</td><td>List of IE8-specific CSS files to always be loaded for the agent interface.</td></tr>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Core::Web</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>1</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'Loader::Agent::CommonCSS::IE8'}->{'000-Framework'} = [
  'Core.OverviewSmall.IE8.css'
];
</pre></td></tr>
</table>

2.29.30. Loader::Agent::CommonJS###000-Framework

<table>
<tr><td>Description:</td><td>List of JS files to always be loaded for the agent interface.</td></tr>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Core::Web</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>1</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'Loader::Agent::CommonJS'}->{'000-Framework'} = [
  'thirdparty/jquery-1.10.0/jquery.js',
  'thirdparty/jquery-browser-detection/jquery-browser-detection.js',
  'thirdparty/jquery-ui-1.10.3/jquery-ui.js',
  'thirdparty/jquery-validate-1.11.1/jquery.validate.js',
  'thirdparty/stacktrace-0.4/stacktrace.js',
  'thirdparty/jquery-pubsub/pubsub.js',
  'thirdparty/jquery-jstree-v.pre1.0/jquery.jstree.js',
  'thirdparty/jquery-jstree-v.pre1.0/_lib/jquery.hotkeys.js',
  'Core.JavaScriptEnhancements.js',
  'Core.Debug.js',
  'Core.Data.js',
  'Core.Config.js',
  'Core.Exception.js',
  'Core.JSON.js',
  'Core.AJAX.js',
  'Core.App.js',
  'Core.UI.js',
  'Core.UI.Accordion.js',
  'Core.UI.Datepicker.js',
  'Core.UI.DnD.js',
  'Core.UI.Resizable.js',
  'Core.UI.Table.js',
  'Core.UI.Accessibility.js',
  'Core.UI.RichTextEditor.js',
  'Core.UI.Dialog.js',
  'Core.UI.ActionRow.js',
</pre></td></tr>
</table>

	'Core.UI.Popup.js', 'Core.UI.TreeSelection.js', 'Core.UI.Autocomplete.js', 'Core.Form.js', 'Core.Form.ErrorTooltips.js', 'Core.Form.Validate.js', 'Core.Agent.js', 'Core.Agent.Search.js', 'Core.Agent.CustomerInformationCenterSearch.js'];

2.29.31. Loader::Agent::CommonJS###001-JQueryMigrate

Description:	List of JS files to always be loaded for the agent interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	$Self->{'Loader::Agent::CommonJS'}->{'001-JQueryMigrate'} = ['thirdparty/jquery-migrate-1.2.1/jquery-migrate.js'];

2.29.32. Loader::Customer::CommonCSS###000-Framework

Description:	List of CSS files to always be loaded for the customer interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	$Self->{'Loader::Customer::CommonCSS'}->{'000-Framework'} = ['Core.Reset.css', 'Core.Default.css', 'Core.Form.css', 'Core.Dialog.css', 'Core.Tooltip.css', 'Core.Login.css', 'Core.Control.css', 'Core.Table.css', 'Core.TicketZoom.css', 'Core.Print.css', 'thirdparty/fontawesome/font-awesome.css'];

2.29.33. Loader::Customer::CommonCSS::IE8###000-Framework

Description:	List of IE8-specific CSS files to always be loaded for the customer interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	$Self->{'Loader::Customer::CommonCSS::IE8'}->{'000-Framework'} = [];

2.29.34. Loader::Customer::CommonJS###000-Framework

Description:	List of JS files to always be loaded for the customer interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Loader::Customer::CommonJS'}->{'000-Framework'} =  [`
	`  'thirdparty/jquery-1.10.0/jquery.js',`
	`  'thirdparty/jquery-browser-detection/jquery-browser-detection.js',`
	`  'thirdparty/jquery-validate-1.11.1/jquery.validate.js',`
	`  'thirdparty/jquery-ui-1.10.3/jquery-ui.js',`
	`  'thirdparty/stacktrace-0.4/stacktrace.js',`
	`  'thirdparty/jquery-pubsub/pubsub.js',`
	`  'thirdparty/jquery-jstree-v.pre1.0/jquery.jstree.js',`
	`  'thirdparty/jquery-jstree-v.pre1.0/_lib/jquery.hotkeys.js',`
	`  'Core.Debug.js',`
	`  'Core.Data.js',`
	`  'Core.Exception.js',`
	`  'Core.JSON.js',`
	`  'Core.JavaScriptEnhancements.js',`
	`  'Core.Config.js',`
	`  'Core.AJAX.js',`
	`  'Core.App.js',`
	`  'Core.UI.js',`
	`  'Core.UI.Accessibility.js',`
	`  'Core.UI.Dialog.js',`
	`  'Core.UI.RichTextEditor.js',`
	`  'Core.UI.Datepicker.js',`
	`  'Core.UI.Popup.js',`
	`  'Core.UI.TreeSelection.js',`
	`  'Core.UI.Autocomplete.js',`
	`  'Core.Form.js',`
	`  'Core.Form.ErrorTooltips.js',`
	`  'Core.Form.Validate.js',`
	`  'Core.Customer.js'`
	`];`

2.29.35. Loader::Customer::CommonJS###001-JQueryMigrate

Description:	List of JS files to always be loaded for the agent interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Loader::Customer::CommonJS'}->{'001-JQueryMigrate'} =  [`
	`  'thirdparty/jquery-migrate-1.2.1/jquery-migrate.js'`
	`];`

2.29.36. Loader::Agent::DefaultSelectedSkin

Description:	The agent skin's InternalName which should be used in the agent interface. Please check the available skins in Frontend::Agent::Skins.
Group:	Framework
SubGroup:	Core::Web

Valid:	1
Required:	1
Config-Setting:	`$Self->{'Loader::Agent::DefaultSelectedSkin'} =  'default';`

2.29.37. Loader::Customer::SelectedSkin::HostBased

Description:	It is possible to configure different skins, for example to distinguish between diferent customers, to be used on a per-domain basis within the application. Using a regular expression (regex), you can configure a Key/Content pair to match a domain. The value in "Key" should match the domain, and the value in "Content" should be a valid skin on your system. Please see the example entries for the proper form of the regex.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Loader::Customer::SelectedSkin::HostBased'} =  {`
	`  'host1\\.example\\.com' => 'Someskin1',`
	`  'host2\\.example\\.com' => 'Someskin2'`
	`};`

2.30. Core::WebUserAgent

2.30.1. WebUserAgent::Timeout

Description:	Sets the timeout (in seconds) for http/ftp downloads.
Group:	Framework
SubGroup:	Core::WebUserAgent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'WebUserAgent::Timeout'} =  '15';`

2.30.2. WebUserAgent::Proxy

Description:	Defines the connections for http/ftp, via a proxy.
Group:	Framework
SubGroup:	Core::WebUserAgent
Valid:	0
Required:	0
Config-Setting:	`$Self->{'WebUserAgent::Proxy'} =  'http://proxy.sn.no:8001/';`

2.31. Crypt::PGP

2.31.1. PGP

Description:	Enables PGP support. When PGP support is enabled for signing and encrypting mail, it is HIGHLY recommended that the web server

	runs 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:	`$Self->{'PGP'} =  '0';`

2.31.2. PGP::Bin

Description:	Defines the path to PGP binary.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PGP::Bin'} =  '/usr/bin/gpg';`

2.31.3. PGP::Options

Description:	Sets the options for PGP binary.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PGP::Options'} =  '--homedir /opt/otrs/.gnupg/ --batch --no-tty --yes';`

2.31.4. PGP::Key::Password

Description:	Sets the password for private PGP key.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1

Config-Setting:	`$Self->{'PGP::Key::Password'} =  {`
	`  '488A0B8F' => 'SomePassword',`
	`  'D2DF79FA' => 'SomePassword'`
	`};`

2.31.5. PGP::TrustedNetwork

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:	`$Self->{'PGP::TrustedNetwork'} =  '0';`

2.31.6. PGP::Log

Description:	Configure your own log text for PGP.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1

Config-Setting:	`$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.',`
	` '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.'`
	`};`

2.31.7. PGP::StoreDecryptedData

Description:	If this option is enabled, then the decrypted data will be stored in the database if they are displayed in AgentTicketZoom.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PGP::StoreDecryptedData'} =  '1';`

2.32. Crypt::SMIME

2.32.1. SMIME

Description:	Enables S/MIME support.
Group:	Framework
SubGroup:	Crypt::SMIME
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SMIME'} =  '0';`

2.32.2. SMIME::Bin

Description:	Defines the path to open ssl binary. It may need a HOME env ($ENV{HOME} = '/var/lib/wwwrun';).
Group:	Framework
SubGroup:	Crypt::SMIME
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SMIME::Bin'} =  '/usr/bin/openssl';`

2.32.3. SMIME::CertPath

Description:	Specifies the directory where SSL certificates are stored.
Group:	Framework
SubGroup:	Crypt::SMIME
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SMIME::CertPath'} =  '/etc/ssl/certs';`

2.32.4. SMIME::PrivatePath

Description:	Specifies the directory where private SSL certificates are stored.
Group:	Framework
SubGroup:	Crypt::SMIME
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SMIME::PrivatePath'} =  '/etc/ssl/private';`

2.32.5. SMIME::CacheTTL

Description:	Cache time in seconds for the SSL certificate attributes.
Group:	Framework
SubGroup:	Crypt::SMIME
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SMIME::CacheTTL'} =  '86400';`

2.32.6. SMIME::StoreDecryptedData

Description:	If this option is enabled, then the decrypted data will be stored in the database if they are displayed in AgentTicketZoom.
Group:	Framework
SubGroup:	Crypt::SMIME
Valid:	1
Required:	1

Config-Setting:	`$Self->{'SMIME::StoreDecryptedData'} =  '1';`

2.33. CustomerInformationCenter

2.33.1. AgentCustomerInformationCenter::MainMenu###010-EditCustomerID

Description:	Main menu registration.
Group:	Framework
SubGroup:	CustomerInformationCenter
Valid:	0
Required:	0

Config-Setting:	`$Self->{'AgentCustomerInformationCenter::MainMenu'}->{'010-EditCustomerID'} =  {`
	`  'Link' => '[% Env("Baselink") %]Action=AdminCustomerCompany;Subaction=Change;CustomerID=[% Data.CustomerID | uri %];Nav=0',`
	`  'Name' => 'Edit customer company'`
	`};`

2.34. Frontend::Admin

2.34.1. Events###Package

Description:	List of all Package events to be displayed in the GUI.
Group:	Framework
SubGroup:	Frontend::Admin
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Events'}->{'Package'} =  [`
	`  'PackageInstall',`
	`  'PackageReinstall',`
	`  'PackageUpgrade',`
	`  'PackageUninstall'`
	`];`

2.34.2. Events###DynamicField

Description:	List of all DynamicField events to be displayed in the GUI.
Group:	Framework
SubGroup:	Frontend::Admin
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Events'}->{'DynamicField'} =  [`
	`  'DynamicFieldAdd',`
	`  'DynamicFieldUpdate',`
	`  'DynamicFieldDelete'`
	`];`

2.34.3. Events###CustomerUser

Description:	List of all CustomerUser events to be displayed in the GUI.
Group:	Framework

SubGroup:	Frontend::Admin
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Events'}->{'CustomerUser'} =  [`
	`  'CustomerUserAdd',`
	`  'CustomerUserUpdate'`
	`];`

2.34.4. Events###CustomerCompany

Description:	List of all CustomerCompany events to be displayed in the GUI.
Group:	Framework
SubGroup:	Frontend::Admin
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Events'}->{'CustomerCompany'} =  [`
	`  'CustomerCompanyAdd',`
	`  'CustomerCompanyUpdate'`
	`];`

2.35. Frontend::Admin::AdminCustomerUser

2.35.1. AdminCustomerUser::RunInitialWildcardSearch

Description:	Runs an initial wildcard search of the existing customer users when accessing the AdminCustomerUser module.
Group:	Framework
SubGroup:	Frontend::Admin::AdminCustomerUser
Valid:	1
Required:	1
Config-Setting:	`$Self->{'AdminCustomerUser::RunInitialWildcardSearch'} =  '1';`

2.36. Frontend::Admin::ModuleRegistration

2.36.1. Frontend::Module###Admin

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'Admin'} =  {`
	`  'Description' => 'Admin-Area',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'Loader' => {`
	`    'CSS' => [`
	`      'Core.Agent.Admin.css'`
	`    ],`
	`    'JavaScript' => [`

```
      'Core.Agent.Admin.SysConfig.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'a',
      'Block' => 'ItemArea',
      'Description' => '',
      'Link' => 'Action=Admin',
      'LinkOption' => '',
      'Name' => 'Admin',
      'NavBar' => 'Admin',
      'Prio' => '10000',
      'Type' => 'Menu'
    }
  ],
  'NavBarModule' => {
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin'
  },
  'NavBarName' => 'Admin',
  'Title' => ''
};
```

2.36.2. Frontend::Module###AdminInit

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
$Self->{'Frontend::Module'}->{'AdminInit'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarName' => '',
  'Title' => 'Init'
};
```

2.36.3. Frontend::Module###AdminUser

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
$Self->{'Frontend::Module'}->{'AdminUser'} =  {
  'Description' => 'Create and manage agents.',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Agent',
    'Description' => 'Create and manage agents.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Agents',
    'Prio' => '100'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Agents'
};
```

2.36.4. Frontend::Module###AdminGroup

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminGroup'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Agent',
    'Description' => 'Create and manage groups.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Groups',
    'Prio' => '150'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Groups'
};
```

2.36.5. Frontend::Module###AdminUserGroup

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminUserGroup'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Agent',
    'Description' => 'Link agents to groups.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Agents <-> Groups',
    'Prio' => '200'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Agents <-> Groups'
};
```

2.36.6. Frontend::Module###AdminCustomerUser

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration

Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AdminCustomerUser'} =  {
  'Description' => 'Edit Customer Users',
  'Group' => [
    'admin',
    'users'
  ],
  'GroupRo' => [
    ''
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketAction.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => '',
      'Block' => 'ItemArea',
      'Description' => '',
      'Link' => 'Action=AdminCustomerUser;Nav=Agent',
      'LinkOption' => '',
      'Name' => 'Customer User Administration',
      'NavBar' => 'Customers',
      'Prio' => '9000',
      'Type' => ''
    }
  ],
  'NavBarModule' => {
    'Block' => 'Customer',
    'Description' => 'Create and manage customer users.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Customer User',
    'Prio' => '300'
  },
  'NavBarName' => 'Customers',
  'Title' => 'Customer Users'
};
```

2.36.7. Frontend::Module###AdminCustomerCompany

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AdminCustomerCompany'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin',
    'users'
  ],
  'GroupRo' => [
    ''
  ],
  'NavBar' => [
    {
      'AccessKey' => '',
      'Block' => 'ItemArea',
      'Description' => '',
      'Link' => 'Action=AdminCustomerCompany;Nav=Agent',
      'LinkOption' => '',
      'Name' => 'Customer Administration',
      'NavBar' => 'Customers',
      'Prio' => '9100',
      'Type' => ''
    }
  ],
  'NavBarModule' => {
    'Block' => 'Customer',
    'Description' => 'Create and manage customers.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Customers',
    'Prio' => '310'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Customer Companies'
};
```

2.36.8. Frontend::Module###AdminCustomerUserGroup

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminCustomerUserGroup'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Customer',
    'Description' => 'Link customer user to groups.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Customer User <-> Groups',
    'Prio' => '400'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Customers <-> Groups'
};
```

2.36.9. Frontend::Module###AdminCustomerUserService

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminCustomerUserService'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Customer',
    'Description' => 'Link customer user to services.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Customer User <-> Services',
    'Prio' => '500'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Customer User <-> Services'
};
```

2.36.10. Frontend::Module###AdminRole

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminRole'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Agent',
    'Description' => 'Create and manage roles.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Roles',
    'Prio' => '600'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Roles'
};
```

2.36.11. Frontend::Module###AdminRoleUser

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminRoleUser'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Agent',
    'Description' => 'Link agents to roles.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Agents <-> Roles',
    'Prio' => '700'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Agents <-> Roles'
};
```

2.36.12. Frontend::Module###AdminRoleGroup

Description:	Frontend module registration for the agent interface.

Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
 $Self->{'Frontend::Module'}->{'AdminRoleGroup'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Agent',
    'Description' => 'Link roles to groups.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Roles <-> Groups',
    'Prio' => '800'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Roles <-> Groups'
};
```

2.36.13. Frontend::Module###AdminSMIME

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
 $Self->{'Frontend::Module'}->{'AdminSMIME'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Email',
    'Description' => 'Manage S/MIME certificates for email
 encryption.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'S/MIME Certificates',
    'Prio' => '1100'
  },
  'NavBarName' => 'Admin',
  'Title' => 'S/MIME Management'
};
```

2.36.14. Frontend::Module###AdminPGP

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
 $Self->{'Frontend::Module'}->{'AdminPGP'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Email',
    'Description' => 'Manage PGP keys for email encryption.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'PGP Keys',
    'Prio' => '1200'
  },
  'NavBarName' => 'Admin',
  'Title' => 'PGP Key Management'
};
```

2.36.15. Frontend::Module###AdminMailAccount

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminMailAccount'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Email',
    'Description' => 'Manage POP3 or IMAP accounts to fetch email
 from.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'PostMaster Mail Accounts',
    'Prio' => '100'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Mail Accounts'
};
```

2.36.16. Frontend::Module###AdminPostMasterFilter

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminPostMasterFilter'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Email',
    'Description' => 'Filter incoming emails.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'PostMaster Filters',
    'Prio' => '200'
  },
  'NavBarName' => 'Admin',
  'Title' => 'PostMaster Filters'
};
```

2.36.17. Frontend::Module###AdminEmail

<table>
<tr><td>Description:</td><td>Frontend module registration for the agent interface.</td></tr>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Admin::ModuleRegistration</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'Frontend::Module'}->{'AdminEmail'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Send notifications to users.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Admin Notification',
    'Prio' => '400'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Admin Notification'
};
</pre></td></tr>
</table>

2.36.18. Frontend::Module###AdminSession

<table>
<tr><td>Description:</td><td>Frontend module registration for the agent interface.</td></tr>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Admin::ModuleRegistration</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'Frontend::Module'}->{'AdminSession'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Manage existing sessions.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Session Management',
    'Prio' => '500'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Session Management'
};
</pre></td></tr>
</table>

2.36.19. Frontend::Module###AdminPerformanceLog

<table>
<tr><td>Description:</td><td>Frontend module registration for the agent interface.</td></tr>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Admin::ModuleRegistration</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'Frontend::Module'}->{'AdminPerformanceLog'} = {
</pre></td></tr>
</table>

```
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.PerformanceLog.css'
    ]
  },
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'View performance benchmark results.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Performance Log',
    'Prio' => '550'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Performance Log'
};
```

2.36.20. Frontend::Module###AdminRegistration

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
 $Self->{'Frontend::Module'}->{'AdminRegistration'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.Registration.css'
    ]
  },
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Manage system registration.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'System Registration',
    'Prio' => '350'
  },
  'NavBarName' => 'Admin',
  'Title' => 'System Registration'
};
```

2.36.21. Frontend::Module###AdminServiceCenter

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
 $Self->{'Frontend::Module'}->{'AdminServiceCenter'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ServiceCenter.css'
    ]
  },
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Manage OTRS Group services.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Service Center',
    'Prio' => '351'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Service Center'
};
```

2.36.22. Frontend::Module###AdminLog

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminLog'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'View system log messages.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'System Log',
    'Prio' => '600'
  },
  'NavBarName' => 'Admin',
  'Title' => 'System Log'
};
```

2.36.23. Frontend::Module###AdminSelectBox

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminSelectBox'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Execute SQL statements.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'SQL Box',
```

	'Prio' => '700'
	  },
	  'NavBarName' => 'Admin',
	  'Title' => 'SQL Box'
	};

2.36.24. Frontend::Module###AdminPackageManager

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AdminPackageManager'} = {
	  'Description' => 'Software Package Manager',
	  'Group' => [
	    'admin'
	  ],
	  'NavBarModule' => {
	    'Block' => 'System',
	    'Description' => 'Update and extend your system with software packages.',
	    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	    'Name' => 'Package Manager',
	    'Prio' => '1000'
	  },
	  'NavBarName' => 'Admin',
	  'Title' => 'Package Manager'
	};

2.37. Frontend::Agent

2.37.1. AgentLogo

Description:	The logo shown in the header of the agent interface. The URL to the image can be a relative URL to the skin image directory, or a full URL to a remote web server.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0

Config-Setting:	$Self->{'AgentLogo'} = {
	  'StyleHeight' => '67px',
	  'StyleRight' => '38px',
	  'StyleTop' => '-4px',
	  'StyleWidth' => '244px',
	  'URL' => 'skins/Agent/default/img/logo_bg.png'
	};

2.37.2. AgentLoginLogo

Description:	The logo shown on top of the login box of the agent interface. The URL to the image must be relative URL to the skin image directory.
Group:	Framework
SubGroup:	Frontend::Agent

Valid:	0
Required:	0

Config-Setting:	`$Self->{'AgentLoginLogo'} =  {`
	`'StyleHeight' => '100px',`
	`'URL' => 'skins/Agent/default/img/loginlogo_default.png'`
	`};`

2.37.3. LoginURL

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

2.37.4. LogoutURL

Description:	Defines an alternate URL, where the logout link refers to.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	`$Self->{'LogoutURL'} =  'http://host.example.com/thanks-for-using-otrs.html';`

2.37.5. PreApplicationModule###AgentInfo

Description:	Defines a useful module to load specific user options or to display news.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0

Config-Setting:	`$Self->{'PreApplicationModule'}->{'AgentInfo'} =`
	`'Kernel::Modules::AgentInfo';`

2.37.6. InfoKey

Description:	Defines the key to be checked with Kernel::Modules::AgentInfo module. If this user preferences key is true, the message is accepted by the system.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'InfoKey'} =  'wpt22';`

2.37.7. InfoFile

Description:	File that is displayed in the Kernel::Modules::AgentInfo module, if located under Kernel/Output/HTML/Standard/AgentInfo.dtl.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'InfoFile'} =  'AgentInfo';`

2.37.8. LostPassword

Description:	Activates lost password feature for agents, in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'LostPassword'} =  '1';`

2.37.9. ShowMotd

Description:	Shows the message of the day on login screen of the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'ShowMotd'} =  '0';`

2.37.10. NotificationSubjectLostPasswordToken

Description:	Defines the subject for notification mails sent to agents, with token about new requested password.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'NotificationSubjectLostPasswordToken'} =  'New OTRS password request';`

2.37.11. NotificationBodyLostPasswordToken

Description:	Defines the body text for notification mails sent to agents, with token about new requested password (after using this link the new password will be sent).
Group:	Framework

SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	$Self->{'NotificationBodyLostPasswordToken'} = 'Hi <OTRS_USERFIRSTNAME>, You or someone impersonating you has requested to change your OTRS password. If you want to do this, click on the link below. You will receive another email containing the password. <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/ <OTRS_CONFIG_ScriptAlias>index.pl? Action=LostPassword;Token=<OTRS_TOKEN> If you did not request a new password, please ignore this email. ';

2.37.12. NotificationSubjectLostPassword

Description:	Defines the subject for notification mails sent to agents, about new password.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	$Self->{'NotificationSubjectLostPassword'} = 'New OTRS password';

2.37.13. NotificationBodyLostPassword

Description:	Defines the body text for notification mails sent to agents, about new password (after using this link the new password will be sent).
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	$Self->{'NotificationBodyLostPassword'} = 'Hi <OTRS_USERFIRSTNAME>, Here\'s your new OTRS password. New password: <OTRS_NEWPW> You can log in via the following URL: <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/ <OTRS_CONFIG_ScriptAlias>index.pl ';

2.37.14. OpenMainMenuOnHover

Description:	If enabled, the first level of the main menu opens on mouse hover (instead of click only).

Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'OpenMainMenuOnHover'} =  '0';`

2.37.15. FirstnameLastnameOrder

Description:	Specifies the order in which the firstname and the lastname of agents will be displayed.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'FirstnameLastnameOrder'} =  '0';`

2.37.16. Loader::Agent::Skin###000-default

Description:	Default skin for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Loader::Agent::Skin'}->{'000-default'} =  {`
	`  'Description' => 'This is the default orange - black skin.',`
	`  'HomePage' => 'www.otrs.org',`
	`  'InternalName' => 'default',`
	`  'VisibleName' => 'Default'`
	`};`

2.37.17. Loader::Agent::Skin###001-slim

Description:	Default skin for the agent interface (slim version).
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Loader::Agent::Skin'}->{'001-slim'} =  {`
	`  'Description' => '"Slim" skin which tries to save screen space for power users.',`
	`  'HomePage' => 'www.otrs.org',`
	`  'InternalName' => 'slim',`
	`  'VisibleName' => 'Default (Slim)'`
	`};`

2.37.18. Loader::Agent::Skin###001-ivory

Description:	Balanced white skin by Felix Niklas.

Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Loader::Agent::Skin'}->{'001-ivory'} =  {`
	`  'Description' => 'Balanced white skin by Felix Niklas.',`
	`  'HomePage' => 'www.felixniklas.de',`
	`  'InternalName' => 'ivory',`
	`  'VisibleName' => 'Ivory'`
	`};`

2.37.19. Loader::Agent::Skin###001-ivory-slim

Description:	Balanced white skin by Felix Niklas (slim version).
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Loader::Agent::Skin'}->{'001-ivory-slim'} =  {`
	`  'Description' => 'Balanced white skin by Felix Niklas (slim version).',`
	`  'HomePage' => 'www.felixniklas.de',`
	`  'InternalName' => 'ivory-slim',`
	`  'VisibleName' => 'Ivory (Slim)'`
	`};`

2.37.20. Loader::Agent::DefaultSelectedSkin::HostBased

Description:	It is possible to configure different skins, for example to distinguish between diferent agents, to be used on a per-domain basis within the application. Using a regular expression (regex), you can configure a Key/Content pair to match a domain. The value in "Key" should match the domain, and the value in "Content" should be a valid skin on your system. Please see the example entries for the proper form of the regex.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Loader::Agent::DefaultSelectedSkin::HostBased'} =  {`
	`  'host1\\.example\\.com' => 'SomeSkin1',`
	`  'host2\\.example\\.com' => 'SomeSkin2'`
	`};`

2.37.21. AutoComplete::Agent###Default

Description:	Defines the config options for the autocompletion feature.
Group:	Framework
SubGroup:	Frontend::Agent

Valid:	1
Required:	0

Config-Setting:	$Self->{'AutoComplete::Agent'}->{'Default'} = {
	  'AutoCompleteActive' => '1',
	  'ButtonText' => 'Search',
	  'MaxResultsDisplayed' => '20',
	  'MinQueryLength' => '2',
	  'QueryDelay' => '100'
	};

2.37.22. AutoComplete::Agent###CustomerSearch

Description:	Defines the config options for the autocompletion feature.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0

Config-Setting:	$Self->{'AutoComplete::Agent'}->{'CustomerSearch'} = {
	  'AutoCompleteActive' => '1',
	  'ButtonText' => 'Search Customer',
	  'MaxResultsDisplayed' => '20',
	  'MinQueryLength' => '2',
	  'QueryDelay' => '100'
	};

2.37.23. AutoComplete::Agent###UserSearch

Description:	Defines the config options for the autocompletion feature.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0

Config-Setting:	$Self->{'AutoComplete::Agent'}->{'UserSearch'} = {
	  'AutoCompleteActive' => '1',
	  'ButtonText' => 'Search User',
	  'MaxResultsDisplayed' => '20',
	  'MinQueryLength' => '2',
	  'QueryDelay' => '100'
	};

2.37.24. PossibleNextActions

Description:	Defines the list of possible next actions on an error screen.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0

Config-Setting:	$Self->{'PossibleNextActions'} = {
	  'Go to dashboard!' => '[% Env("CGIHandle") %]?Action=AgentDashboard'
	};

2.38. Frontend::Agent::Dashboard

2.38.1. AgentCustomerInformationCenter::Backend###0600-CIC-CustomerCompanyInformation

Description:	Parameters for the dashboard backend of the customer company information of the agent interface . "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	`$Self->{'AgentCustomerInformationCenter::Backend'}->{'0600-CIC-CustomerCompanyInformation'} =  {`
	`  'Attributes' => '',`
	`  'Block' => 'ContentSmall',`
	`  'Default' => '1',`
	`  'Description' => 'Customer Information',`
	`  'Group' => '',`
	`  'Module' => 'Kernel::Output::HTML::DashboardCustomerCompanyInformation',`
	`  'Title' => 'Customer Information'`
	`};`

2.38.2. DashboardBackend###0000-ProductNotify

Description:	Defines the parameters for the dashboard backend. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" defines the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	`$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'`
	`};`

2.38.3. DashboardBackend###0390-UserOutOfOffice

Description:	Defines the parameters for the dashboard backend. "Limit" defines the number of entries displayed by default. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to

	enable it manually. "CacheTTLLocal" defines the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	$Self->{'DashboardBackend'}->{'0390-UserOutOfOffice'} = {
	  'Block' => 'ContentSmall',
	  'CacheTTLLocal' => '5',
	  'Default' => '1',
	  'Description' => '',
	  'Group' => '',
	  'IdleMinutes' => '60',
	  'Limit' => '10',
	  'Module' => 'Kernel::Output::HTML::DashboardUserOutOfOffice',
	  'SortBy' => 'UserFullname',
	  'Title' => 'Out Of Office'
	};

2.38.4. DashboardBackend###0400-UserOnline

Description:	Defines the parameters for the dashboard backend. "Limit" defines the number of entries displayed by default. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" defines the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	$Self->{'DashboardBackend'}->{'0400-UserOnline'} = {
	  'Block' => 'ContentSmall',
	  'CacheTTLLocal' => '5',
	  'Default' => '0',
	  'Description' => '',
	  'Filter' => 'Agent',
	  'Group' => '',
	  'IdleMinutes' => '60',
	  'Limit' => '10',
	  'Module' => 'Kernel::Output::HTML::DashboardUserOnline',
	  'ShowEmail' => '0',
	  'SortBy' => 'UserFullname',
	  'Title' => 'Online'
	};

2.38.5. DashboardBackend###0410-RSS

Description:	Defines the parameters for the dashboard backend. "Limit" defines the number of entries displayed by default. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTL" indicates the cache expiration period in minutes for the plugin.
Group:	Framework

SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DashboardBackend'}->{'0410-RSS'} =  {`
	`  'Block' => 'ContentSmall',`
	`  'CacheTTL' => '360',`
	`  'Default' => '1',`
	`  'Description' => '',`
	`  'Group' => '',`
	`  'Limit' => '6',`
	`  'Module' => 'Kernel::Output::HTML::DashboardRSS',`
	`  'Title' => 'OTRS News'`
	`};`

2.38.6. DashboardBackend###0200-Image

Description:	Defines the parameters for the dashboard backend. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTL" indicates the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	0
Required:	0

Config-Setting:	`$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'`
	`};`

2.38.7. DashboardBackend###0210-MOTD

Description:	Shows the message of the day (MOTD) in the agent dashboard. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	0
Required:	0

Config-Setting:	`$Self->{'DashboardBackend'}->{'0210-MOTD'} =  {`
	`  'Block' => 'ContentLarge',`
	`  'Default' => '1',`
	`  'Group' => '',`
	`  'Module' => 'Kernel::Output::HTML::DashboardMOTD',`

	'Title' => 'Message of the Day' };

2.38.8. DashboardBackend###0300-IFrame

Description:	Defines the parameters for the dashboard backend. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTL" indicates the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	0
Required:	0

Config-Setting:	$Self->{'DashboardBackend'}->{'0300-IFrame'} = {
	  'Align' => 'left',
	  'Block' => 'ContentLarge',
	  'Default' => '1',
	  'Description' => 'Some description!',
	  'Frameborder' => '1',
	  'Group' => '',
	  'Height' => '800',
	  'Link' => 'http://otrs.org/',
	  'LinkTitle' => 'OTRS.org/',
	  'Marginheight' => '5',
	  'Marginwidth' => '5',
	  'Module' => 'Kernel::Output::HTML::DashboardIFrame',
	  'Scrolling' => 'auto',
	  'Title' => 'A Website',
	  'URL' => 'http://www.otrs.org/',
	  'Width' => '1024'
	};

2.38.9. AgentCustomerInformationCenter::Backend###0050-CIC-CustomerUserList

Description:	Parameters for the dashboard backend of the customer user list overview of the agent interface . "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	$Self->{'AgentCustomerInformationCenter::Backend'}->{'0050-CIC-CustomerUserList'} = {
	  'Attributes' => '',
	  'Block' => 'ContentLarge',
	  'CacheTTLLocal' => '0.5',
	  'Default' => '1',
	  'Description' => 'All customer users of a CustomerID',
	  'Group' => '',
	  'Limit' => '10',
	  'Module' => 'Kernel::Output::HTML::DashboardCustomerUserList',

	`  'Permission' => 'ro',`
	`  'Title' => 'Customer Users'`
	`};`

2.39. Frontend::Agent::LinkObject

2.39.1. Frontend::AgentLinkObject::WildcardSearch

Description:	Starts a wildcard search of the active object after the link object mask is started.
Group:	Framework
SubGroup:	Frontend::Agent::LinkObject
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::AgentLinkObject::WildcardSearch'} =  '0';`

2.40. Frontend::Agent::ModuleMetaHead

2.40.1. Frontend::HeaderMetaModule###100-Refresh

Description:	Defines the module to generate html refresh headers of html sites.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleMetaHead
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Frontend::HeaderMetaModule'}->{'100-Refresh'} =  {`
	`  'Module' => 'Kernel::Output::HTML::HeaderMetaRefresh'`
	`};`

2.41. Frontend::Agent::ModuleNotify

2.41.1. Frontend::NotifyModule###200-UID-Check

Description:	Defines the module to display a notification in the agent interface, if the system is used by the admin user (normally you shouldn't work as admin).
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'200-UID-Check'} =  {`
	`  'Module' => 'Kernel::Output::HTML::NotificationUIDCheck'`
	`};`

2.41.2. Frontend::NotifyModule###300-ShowAgentOnline

Description:	Defines the module that shows all the currently logged in agents in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify

Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'300-ShowAgentOnline'} =  {`
	`  'IdleMinutes' => '60',`
	`  'Module' => 'Kernel::Output::HTML::NotificationAgentOnline',`
	`  'ShowEmail' => '1'`
	`};`

2.41.3. Frontend::NotifyModule###400-ShowCustomerOnline

Description:	Defines the module that shows all the currently loged in customers in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'400-ShowCustomerOnline'} =  {`
	`  'IdleMinutes' => '60',`
	`  'Module' => 'Kernel::Output::HTML::NotificationCustomerOnline',`
	`  'ShowEmail' => '1'`
	`};`

2.41.4. Frontend::NotifyModule###500-OutofOffice-Check

Description:	Defines the module to display a notification in the agent interface, if the agent is logged in while having out-of-office active.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'500-OutofOffice-Check'} =  {`
	`  'Module' => 'Kernel::Output::HTML::NotificationOutofOfficeCheck'`
	`};`

2.41.5. Frontend::NotifyModule###900-Generic

Description:	Defines the module that shows a generic notification in the agent interface. Either "Text" - if configured - or the contents of "File" will be displayed.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'900-Generic'} =  {`
	`  'File' => '<OTRS_CONFIG_Home>/var/notify.txt',`
	`  'Link' => 'http://www.otrs.com',`
	`  'Module' => 'Kernel::Output::HTML::NotificationGeneric',`
	`  'Priority' => 'Warning',`
	`  'Text' => 'The OTRS Website'`
	`};`

2.42. Frontend::Agent::ModuleRegistration

2.42.1. Frontend::Module###Logout

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

Config-Setting:	`$Self->{'Frontend::Module'}->{'Logout'} =  {`
	`  'Description' => 'Logout',`
	`  'NavBarName' => '',`
	`  'Title' => ''`
	`};`

2.42.2. Frontend::Module###AgentDashboard

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

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentDashboard'} =  {`
	`  'Description' => 'Agent Dashboard',`
	`  'Loader' => {`
	`    'CSS' => [`
	`      'Core.Agent.Dashboard.css',`
	`      'Core.AllocationList.css',`
	`      'thirdparty/fullcalendar-1.6.1/fullcalendar.css',`
	`      'thirdparty/d3js/nv.d3.css'`
	`    ],`
	`    'JavaScript' => [`
	`      'thirdparty/flot-0.8.1/excanvas.js',`
	`      'thirdparty/flot-0.8.1/jquery.flot.js',`
	`      'thirdparty/fullcalendar-1.6.1/fullcalendar.min.js',`
	`      'thirdparty/d3js/d3.v3.min.js',`
	`      'thirdparty/d3js/nv.d3.min.js',`
	`      'thirdparty/d3js/models/OTRSmultiBarChart.js',`
	`      'thirdparty/d3js/models/OTRSstackedAreaChart.js',`
	`      'Core.UI.Chart.js',`
	`      'Core.UI.AdvancedChart.js',`
	`      'Core.UI.DnD.js',`
	`      'Core.UI.AllocationList.js',`
	`      'Core.Agent.TableFilters.js',`
	`      'Core.Agent.Dashboard.js'`
	`    ]`
	`  },`
	`  'NavBar' => [`
	`    {`
	`      'AccessKey' => 'd',`
	`      'Block' => 'ItemArea',`
	`      'Description' => '',`
	`      'Link' => 'Action=AgentDashboard',`
	`      'LinkOption' => '',`
	`      'Name' => 'Dashboard',`
	`      'NavBar' => 'Dashboard',`
	`      'Prio' => '50',`
	`      'Type' => 'Menu'`
	`    }`

```
  ],
  'NavBarName' => 'Dashboard',
  'Title' => ''
};
```

2.42.3. Frontend::Module###AgentCustomerInformationCenter

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AgentCustomerInformationCenter'} =  {
  'Description' => 'Customer Information Center',
  'Loader' => {
    'CSS' => [
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'thirdparty/flot-0.8.1/excanvas.js',
      'thirdparty/flot-0.8.1/jquery.flot.js',
      'Core.UI.Chart.js',
      'Core.UI.AllocationList.js',
      'Core.Agent.Dashboard.js',
      'Core.Agent.TableFilters.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'c',
      'Block' => 'ItemArea',
      'Description' => '',
      'Link' => 'Action=AgentCustomerInformationCenter',
      'LinkOption' => 'onclick="window.setTimeout(function()
{Core.Agent.CustomerInformationCenterSearch.OpenSearchDialog();}, 0);
 return false;"',
      'Name' => 'Customer Information Center',
      'NavBar' => 'Customers',
      'Prio' => '50',
      'Type' => ''
    },
    {
      'AccessKey' => '',
      'Block' => 'ItemArea',
      'Description' => '',
      'Link' => 'Action=AgentCustomerInformationCenter',
      'LinkOption' => '',
      'Name' => 'Customers',
      'NavBar' => 'Customers',
      'Prio' => '60',
      'Type' => 'Menu'
    }
  ],
  'NavBarName' => 'Customer Information Center',
  'Title' => ''
};
```

2.42.4. Frontend::Module###AgentCustomerInformationCenterSearch

Description:	Frontend module registration for the agent interface.
Group:	Framework

SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentCustomerInformationCenterSearch'}
	= {
	  'Description' => 'Customer Information Center Search',
	  'Title' => ''
	};

2.42.5. Frontend::Module###AgentPreferences

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

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentPreferences'} = {
	  'Description' => 'Agent Preferences',
	  'NavBarName' => 'Preferences',
	  'Title' => ''
	};

2.42.6. Frontend::Module###PictureUpload

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	1

Config-Setting:	$Self->{'Frontend::Module'}->{'PictureUpload'} = {
	  'Description' => 'Picture upload module',
	  'NavBarName' => 'Ticket',
	  'Title' => 'Picture-Upload'
	};

2.42.7. Frontend::Module###AgentSpelling

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

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentSpelling'} = {
	  'Description' => 'Spell checker',
	  'Loader' => {
	    'JavaScript' => [
	      'Core.Agent.TicketAction.js'
	    ]
	  },
	  'NavBarName' => '',
	  'Title' => 'Spell Checker'

	};

2.42.8. Frontend::Module###SpellingInline

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

Config-Setting:	$Self->{'Frontend::Module'}->{'SpellingInline'} = {
	  'Description' => 'Spell checker',
	  'NavBarName' => '',
	  'Title' => 'Spell Checker'
	};

2.42.9. Frontend::Module###AgentBook

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

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentBook'} = {
	  'Description' => 'Address book of CustomerUser sources',
	  'Loader' => {
	    'JavaScript' => [
	      'Core.Agent.CustomerSearch.js',
	      'Core.Agent.TicketAction.js'
	    ]
	  },
	  'NavBarName' => '',
	  'Title' => 'Address Book'
	};

2.42.10. Frontend::Module###AgentLinkObject

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

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentLinkObject'} = {
	  'Description' => 'Link Object',
	  'NavBarName' => '',
	  'Title' => 'Link Object'
	};

2.42.11. Frontend::Module###AgentInfo

Description:	Frontend module registration for the agent interface.
Group:	Framework

SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentInfo'} =  {`
	`  'Description' => 'Generic Info module',`
	`  'NavBarName' => '',`
	`  'Title' => 'Info'`
	`};`

2.42.12. Frontend::Module###AgentSearch

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

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentSearch'} =  {`
	`  'Description' => 'Global Search Module',`
	`  'NavBarName' => '',`
	`  'Title' => 'Search'`
	`};`

2.42.13. CustomerFrontend::Module###SpellingInline

Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'SpellingInline'} =  {`
	`  'Description' => 'Spell checker',`
	`  'NavBarName' => '',`
	`  'Title' => 'Spell Checker'`
	`};`

2.42.14. Frontend::Module###AgentHTMLReference

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

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentHTMLReference'} =  {`
	`  'Description' => 'HTML Reference',`
	`  'Group' => [`
	`    'users'`
	`  ],`
	`  'GroupRo' => [`
	`    'users'`
	`  ],`
	`  'Loader' => {`

```
    'CSS' => [
      'Core.Agent.HTMLReference.css'
    ]
  },
  'NavBarName' => '',
  'Title' => 'HTML Reference'
};
```

2.42.15. Frontend::Module###AgentStats

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AgentStats'} =  {
  'Description' => 'Stats',
  'Group' => [
    'stats'
  ],
  'GroupRo' => [
    'stats'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Stats.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => '',
      'Block' => 'ItemArea',
      'Description' => '',
      'Link' => 'Action=AgentStats;Subaction=Overview',
      'LinkOption' => '',
      'Name' => 'Statistics',
      'NavBar' => 'Stats',
      'Prio' => '8500',
      'Type' => 'Menu'
    },
    {
      'AccessKey' => '',
      'Block' => '',
      'Description' => 'Overview',
      'GroupRo' => [
        'stats'
      ],
      'Link' => 'Action=AgentStats;Subaction=Overview',
      'LinkOption' => '',
      'Name' => 'Overview',
      'NavBar' => 'Stats',
      'Prio' => '100',
      'Type' => ''
    },
    {
      'AccessKey' => '',
      'Block' => '',
      'Description' => 'New',
      'Group' => [
        'stats'
      ],
      'Link' => 'Action=AgentStats;Subaction=Add',
      'LinkOption' => '',
      'Name' => 'New',
      'NavBar' => 'Stats',
      'Prio' => '200',
      'Type' => ''
    },
    {
      'AccessKey' => '',
      'Block' => '',
      'Description' => 'Import',
      'Group' => [
        'stats'
      ],
      'Link' => 'Action=AgentStats;Subaction=Import',
      'LinkOption' => '',
      'Name' => 'Import',
      'NavBar' => 'Stats',
      'Prio' => '300',
      'Type' => ''
    }
  ],
  'NavBarName' => 'Stats',
  'Title' => 'Stats'
};
```

2.43. Frontend::Agent::NavBarModule

2.43.1. Frontend::NavBarModule###6-CustomerCompany

Description:	Frontend module registration (disable company link if no company feature is used).
Group:	Framework
SubGroup:	Frontend::Agent::NavBarModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::NavBarModule'}->{'6-CustomerCompany'} = {`
	`'Module' => 'Kernel::Output::HTML::NavBarCustomerCompany'`
	`};`

2.43.2. Frontend::NavBarModule###7-AgentTicketService

Description:	Frontend module registration (disable AgentTicketService link if Ticket Serivice feature is not used).
Group:	Framework
SubGroup:	Frontend::Agent::NavBarModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::NavBarModule'}->{'7-AgentTicketService'} = {`
	`'Module' => 'Kernel::Output::HTML::NavBarAgentTicketService'`
	`};`

2.44. Frontend::Agent::Preferences

2.44.1. PreferencesTableValue

Description:	Defines the name of the column to store the data in the preferences table.

Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PreferencesTableValue'} =  'preferences_value';`

2.44.2. PreferencesTableUserID

Description:	Defines the name of the column to store the user identifier in the preferences table.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PreferencesTableUserID'} =  'user_id';`

2.44.3. PreferencesView

Description:	Sets the display order of the different items in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	1

Config-Setting:	`$Self->{'PreferencesView'} =  [`
	`  'User Profile',`
	`  'Email Settings',`
	`  'Other Settings'`
	`];`

2.44.4. PreferencesGroups###Password

Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'Password'} =  {`
	`  'Active' => '1',`
	`  'Area' => 'Agent',`
	`  'Column' => 'User Profile',`
	`  'Label' => 'Change password',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesPassword',`
	`  'PasswordMaxLoginFailed' => '0',`
	`  'PasswordMin2Characters' => '0',`
	`  'PasswordMin2Lower2UpperCharacters' => '0',`
	`  'PasswordMinSize' => '0',`
	`  'PasswordNeedDigit' => '0',`
	`  'PasswordRegExp' => '',`
	`  'Prio' => '0500'`

	};

2.44.5. PreferencesGroups###SpellDict

Description:	Defines the config parameters of this item, to be shown in the preferences view. Take care to maintain the dictionaries installed in the system in the data section.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	0
Required:	0

Config-Setting:	$Self->{'PreferencesGroups'}->{'SpellDict'} = {
	  'Active' => '1',
	  'Column' => 'User Profile',
	  'Data' => {
	    'deutsch' => 'Deutsch',
	    'english' => 'English'
	  },
	  'DataSelected' => 'english',
	  'Key' => 'Default spelling dictionary',
	  'Label' => 'Spelling Dictionary',
	  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	  'PrefKey' => 'UserSpellDict',
	  'Prio' => '2000'
	};

2.44.6. PreferencesGroups###Comment

Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	$Self->{'PreferencesGroups'}->{'Comment'} = {
	  'Active' => '0',
	  'Block' => 'Input',
	  'Column' => 'Other Settings',
	  'Data' => '[% Env("UserComment") %]',
	  'Key' => 'Comment',
	  'Label' => 'Comment',
	  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	  'PrefKey' => 'UserComment',
	  'Prio' => '6000'
	};

2.44.7. PreferencesGroups###Language

Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1

Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'Language'} =  {`
	`  'Active' => '1',`
	`  'Column' => 'User Profile',`
	`  'Key' => 'Frontend language',`
	`  'Label' => 'Language',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesLanguage',`
	`  'PrefKey' => 'UserLanguage',`
	`  'Prio' => '1000'`
	`};`

2.44.8. PreferencesGroups###Skin

Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'Skin'} =  {`
	`  'Active' => '1',`
	`  'Column' => 'User Profile',`
	`  'Key' => 'Wear this frontend skin',`
	`  'Label' => 'Skin',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesSkin',`
	`  'PrefKey' => 'UserSkin',`
	`  'Prio' => '2000'`
	`};`

2.44.9. PreferencesGroups###Theme

Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'Theme'} =  {`
	`  'Active' => '1',`
	`  'Column' => 'User Profile',`
	`  'Key' => 'Frontend theme',`
	`  'Label' => 'Theme',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesTheme',`
	`  'PrefKey' => 'UserTheme',`
	`  'Prio' => '3000'`
	`};`

2.44.10. PreferencesGroups###OutOfOffice

Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences

<table>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td>$Self->{'PreferencesGroups'}->{'OutOfOffice'} = {
  'Active' => '1',
  'Block' => 'OutOfOffice',
  'Column' => 'User Profile',
  'Key' => '',
  'Label' => 'Out Of Office Time',
  'Module' => 'Kernel::Output::HTML::PreferencesOutOfOffice',
  'PrefKey' => 'UserOutOfOffice',
  'Prio' => '4000'
};</td></tr>
</table>

2.44.11. PreferencesGroups###TimeZone

<table>
<tr><td>Description:</td><td>Defines the config parameters of this item, to be shown in the preferences view.</td></tr>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Agent::Preferences</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td>$Self->{'PreferencesGroups'}->{'TimeZone'} = {
  'Active' => '1',
  'Column' => 'User Profile',
  'Key' => 'Time Zone',
  'Label' => 'Time Zone',
  'Module' => 'Kernel::Output::HTML::PreferencesTimeZone',
  'PrefKey' => 'UserTimeZone',
  'Prio' => '5000'
};</td></tr>
</table>

2.44.12. PreferencesGroups###CSVSeparator

<table>
<tr><td>Description:</td><td>Gives end users the possibility to override the separator character for CSV files, defined in the translation files.</td></tr>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Agent::Preferences</td></tr>
<tr><td>Valid:</td><td>0</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td>$Self->{'PreferencesGroups'}->{'CSVSeparator'} = {
  'Active' => '1',
  'Column' => 'Other Settings',
  'Data' => {
    '' => '',
    ',' => ',',
    ';' => ';',
    '\\t' => 'tab',
    '|' => '|'
  },
  'DataSelected' => '0',
  'Desc' => 'Select the separator character used in CSV files (stats and searches). If you don\'t select a separator here, the default separator for your language will be used.',
  'Key' => 'CSV Separator',
  'Label' => 'CSV Separator',
  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',</td></tr>
</table>

	`  'PrefKey' => 'UserCSVSeparator',`
	`  'Prio' => '4000'`
	`};`

2.45. Frontend::Agent::SearchRouter

2.45.1. Frontend::SearchDefault

Description:	Search backend default router.
Group:	Framework
SubGroup:	Frontend::Agent::SearchRouter
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::SearchDefault'} =`
	`'Action=AgentTicketSearch;Subaction=AJAX';`

2.46. Frontend::Agent::Stats

2.46.1. Stats::SearchPageShown

Description:	Defines the default maximum number of search results shown on the overview page.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::SearchPageShown'} =  '20';`

2.46.2. Stats::DefaultSelectedDynamicObject

Description:	Defines the default selection at the drop down menu for dynamic objects (Form: Common Specification).
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::DefaultSelectedDynamicObject'} =  'Ticket';`

2.46.3. Stats::DefaultSelectedPermissions

Description:	Defines the default selection at the drop down menu for permissions (Form: Common Specification).
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Stats::DefaultSelectedPermissions'} =  [`
	`  'stats'`
	`];`

2.46.4. Stats::DefaultSelectedFormat

Description:	Defines the default selection at the drop down menu for stats format (Form: Common Specification). Please insert the format key (see Stats::Format).
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Stats::DefaultSelectedFormat'} =  [`
	`  'Print',`
	`  'CSV'`
	`];`

2.46.5. Stats::SearchLimit

Description:	Defines the search limit for the stats.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::SearchLimit'} =  '500';`

2.46.6. Stats::Format

Description:	Defines all the possible stats output formats.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1

Config-Setting:	`$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'`
	`};`

2.46.7. Stats::GraphSize

Description:	Sets the size of the statistic graph.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Stats::GraphSize'} =  {`
	`  '1200x800' => '1200x800',`
	`  '1600x1200' => '1600x1200',`
	`  '800x600' => '800x600'`
	`};`

2.46.8. Stats::TimeType

Description:	Sets the time type which should be shown.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::TimeType'} =  'Extended';`

2.46.9. Stats::ExchangeAxis

Description:	Allows agents to exchange the axis of a stat if they generate one.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Stats::ExchangeAxis'} =  '0';`

2.46.10. Stats::UseAgentElementInStats

Description:	Allows agents to generate individual-related stats.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Stats::UseAgentElementInStats'} =  '0';`

2.46.11. Stats::UseInvalidAgentInStats

Description:	Allows invalid agents to generate individual-related stats.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Stats::UseInvalidAgentInStats'} =  '1';`

2.46.12. Stats::CustomerIDAsMultiSelect

Description:	Shows all the customer identifiers in a multi-select field (not useful if you have a lot of customer identifiers).
Group:	Framework

SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Stats::CustomerIDAsMultiSelect'} =  '1';`

2.47. Frontend::Customer

2.47.1. CustomerHeadline

Description:	The headline shown in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerHeadline'} =  'Example Company Support';`

2.47.2. CustomerLogo

Description:	The logo shown in the header of the customer interface. The URL to the image can be a relative URL to the skin image directory, or a full URL to a remote web server.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0

Config-Setting:	`$Self->{'CustomerLogo'} =  {`
	`  'StyleHeight' => '50px',`
	`  'StyleRight' => '25px',`
	`  'StyleTop' => '2px',`
	`  'StyleWidth' => '135px',`
	`  'URL' => 'skins/Customer/default/img/logo.png'`
	`};`

2.47.3. CustomerPanelUserID

Description:	Defines the user identifier for the customer panel.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelUserID'} =  '1';`

2.47.4. CustomerGroupSupport

Description:	Activates support for customer groups.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1

Required:	1
Config-Setting:	`$Self->{'CustomerGroupSupport'} =  '0';`

2.47.5. CustomerGroupAlwaysGroups

Description:	Defines the groups every customer user will be in (if CustomerGroupSupport is enabled and you don't want to manage every user for these groups).
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerGroupAlwaysGroups'} =  [`
	`  'users'`
	`];`

2.47.6. CustomerPanelLoginURL

Description:	Defines an alternate login URL for the customer panel..
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'CustomerPanelLoginURL'} =  'http://host.example.com/cgi-bin/login.pl';`

2.47.7. CustomerPanelLogoutURL

Description:	Defines an alternate logout URL for the customer panel.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'CustomerPanelLogoutURL'} =  'http://host.example.com/cgi-bin/login.pl';`

2.47.8. Frontend::CustomerUser::Item###1-GoogleMaps

Description:	Defines a customer item, which generates a google maps icon at the end of a customer info block.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::CustomerUser::Item'}->{'1-GoogleMaps'} =  {`
	`  'Attributes' => 'UserStreet;UserCity;UserCountry;',`
	`  'CSS' => 'Core.Agent.CustomerUser.GoogleMaps.css',`
	`  'CSSClass' => 'GoogleMaps',`

<table>
<tr><td></td><td><pre> 'IconName' => 'fa-globe',
  'Module' => 'Kernel::Output::HTML::CustomerUserGeneric',
  'Required' => 'UserStreet;UserCity;',
  'Target' => '_blank',
  'Text' => 'Location',
  'URL' => 'http://maps.google.com/maps?z=7&q='
};</pre></td></tr>
</table>

2.47.9. Frontend::CustomerUser::Item###2-Google

<table>
<tr><td>Description:</td><td>Defines a customer item, which generates a google icon at the end of a customer info block.</td></tr>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Customer</td></tr>
<tr><td>Valid:</td><td>0</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td><pre> $Self->{'Frontend::CustomerUser::Item'}->{'2-Google'} = {
  'Attributes' => 'UserFirstname;UserLastname;',
  'CSS' => 'Core.Agent.CustomerUser.Google.css',
  'CSSClass' => 'Google',
  'IconName' => 'fa-google',
  'Module' => 'Kernel::Output::HTML::CustomerUserGeneric',
  'Required' => 'UserFirstname;UserLastname;',
  'Target' => '_blank',
  'Text' => 'Google',
  'URL' => 'http://google.com/search?q='
};</pre></td></tr>
</table>

2.47.10. Frontend::CustomerUser::Item###2-LinkedIn

<table>
<tr><td>Description:</td><td>Defines a customer item, which generates a LinkedIn icon at the end of a customer info block.</td></tr>
<tr><td>Group:</td><td>Framework</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Customer</td></tr>
<tr><td>Valid:</td><td>0</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td><pre> $Self->{'Frontend::CustomerUser::Item'}->{'2-LinkedIn'} = {
  'Attributes' => 'UserFirstname;UserLastname;',
  'CSS' => 'Core.Agent.CustomerUser.LinkedIn.css',
  'CSSClass' => 'LinkedIn',
  'IconName' => 'fa-linkedin',
  'Module' => 'Kernel::Output::HTML::CustomerUserGeneric',
  'Required' => 'UserFirstname;UserLastname;',
  'Target' => '_blank',
  'Text' => 'LinkedIn',
  'URL' => 'http://www.linkedin.com/commonSearch?
type=people&keywords='
};</pre></td></tr>
</table>

2.47.11. Frontend::CustomerUser::Item###3-XING

Description:	Defines a customer item, which generates a XING icon at the end of a customer info block.
Group:	Framework
SubGroup:	Frontend::Customer

Valid:	0
Required:	0

Config-Setting:	$Self->{'Frontend::CustomerUser::Item'}->{'3-XING'} = {
	  'Attributes' => 'UserFirstname;UserLastname;',
	  'CSS' => 'Core.Agent.CustomerUser.Xing.css',
	  'CSSClass' => 'Xing',
	  'IconName' => 'fa-xing',
	  'Module' => 'Kernel::Output::HTML::CustomerUserGeneric',
	  'Required' => 'UserFirstname;UserLastname;',
	  'Target' => '_blank',
	  'Text' => 'XING',
	  'URL' => 'https://www.xing.com/app/search?op=search;keywords='
	};

2.47.12. CustomerPanelPreApplicationModule###CustomerAccept

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:	$Self->{'CustomerPanelPreApplicationModule'}->{'CustomerAccept'} = 'Kernel::Modules::CustomerAccept';

2.47.13. CustomerPanel::InfoKey

Description:	Defines the key to check with CustomerAccept. If this user preferences key is true, then the message is accepted by the system.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	$Self->{'CustomerPanel::InfoKey'} = 'CustomerAccept1';

2.47.14. CustomerPanel::InfoFile

Description:	Defines the path of the shown info file, that is located under Kernel/Output/HTML/Standard/CustomerAccept.dtl.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	$Self->{'CustomerPanel::InfoFile'} = 'CustomerAccept';

2.47.15. CustomerPanelLostPassword

Description:	Activates lost password feature for customers.

Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	$Self->{'CustomerPanelLostPassword'} = '1';

2.47.16. CustomerPanelCreateAccount

Description:	Enables customers to create their own accounts.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	$Self->{'CustomerPanelCreateAccount'} = '1';

2.47.17. CustomerPanelSubjectLostPasswordToken

Description:	Defines the subject for notification mails sent to customers, with token about new requested password.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	$Self->{'CustomerPanelSubjectLostPasswordToken'} = 'New OTRS password request';

2.47.18. CustomerPanelBodyLostPasswordToken

Description:	Defines the body text for notification mails sent to customers, with token about new requested password (after using this link the new password will be sent).
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	$Self->{'CustomerPanelBodyLostPasswordToken'} = 'Hi <OTRS_USERFIRSTNAME>, You or someone impersonating you has requested to change your OTRS password. If you want to do this, click on this link. You will receive another email containing the password. <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/<OTRS_CONFIG_ScriptAlias>customer.pl?Action=CustomerLostPassword;Token=<OTRS_TOKEN> If you did not request a new password, please ignore this email. ';

2.47.19. CustomerPanelSubjectLostPassword

Description:	Defines the subject for notification mails sent to customers, about new password.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelSubjectLostPassword'} =  'New OTRS password';`

2.47.20. CustomerPanelBodyLostPassword

Description:	Defines the body text for notification mails sent to customers, about new password (after using this link the new password will be sent).
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	$Self->{'CustomerPanelBodyLostPassword'} = 'Hi <OTRS_USERFIRSTNAME>, New password: <OTRS_NEWPW> <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/ <OTRS_CONFIG_ScriptAlias>customer.pl ';

2.47.21. CustomerPanelSubjectNewAccount

Description:	Defines the subject for notification mails sent to customers, about new account.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelSubjectNewAccount'} =  'New OTRS Account!';`

2.47.22. CustomerPanelBodyNewAccount

Description:	Defines the body text for notification mails sent to customers, about new account.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	$Self->{'CustomerPanelBodyNewAccount'} = 'Hi <OTRS_USERFIRSTNAME>, You or someone impersonating you has created a new OTRS account for you.

	Full name: <OTRS_USERFIRSTNAME> <OTRS_USERLASTNAME> User name: <OTRS_USERLOGIN> Password : <OTRS_USERPASSWORD> You can log in via the following URL. We encourage you to change your password via the Preferences button after logging in. <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/ <OTRS_CONFIG_ScriptAlias>customer.pl ';

2.47.23. Loader::Customer::Skin###000-default

Description:	Default skin for OTRS 3.0 interface.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Loader::Customer::Skin'}->{'000-default'} =  {`
	`  'Description' => 'This is the default orange - black skin for OTRS 3.0.',`
	`  'HomePage' => 'www.otrs.org',`
	`  'InternalName' => 'default',`
	`  'VisibleName' => 'Default'`
	`};`

2.47.24. Loader::Customer::SelectedSkin

Description:	The customer skin's InternalName which should be used in the customer interface. Please check the available skins in Frontend::Customer::Skins.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Loader::Customer::SelectedSkin'} =  'default';`

2.47.25. AutoComplete::Customer###Default

Description:	Defines the config options for the autocompletion feature.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	0

Config-Setting:	`$Self->{'AutoComplete::Customer'}->{'Default'} =  {`
	`  'AutoCompleteActive' => '1',`
	`  'ButtonText' => 'Search',`
	`  'MaxResultsDisplayed' => '20',`
	`  'MinQueryLength' => '2',`
	`  'QueryDelay' => '100'`
	`};`

2.48. Frontend::Customer::Auth

2.48.1. Customer::AuthModule

Description:	Defines the module to authenticate customers.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::DB';`

2.48.2. Customer::AuthModule::DB::CryptType

Description:	If "DB" was selected for Customer::AuthModule, the crypt type of passwords must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Customer::AuthModule::DB::CryptType'} =  'sha2';`

2.48.3. Customer::AuthModule::DB::Table

Description:	If "DB" was selected for Customer::AuthModule, the name of the table where your customer data should be stored must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Customer::AuthModule::DB::Table'} =  'customer_user';`

2.48.4. Customer::AuthModule::DB::CustomerKey

Description:	If "DB" was selected for Customer::AuthModule, the name of the column for the CustomerKey in the customer table must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Customer::AuthModule::DB::CustomerKey'} =  'login';`

2.48.5. Customer::AuthModule::DB::CustomerPassword

Description:	If "DB" was selected for Customer::AuthModule, the column name for the CustomerPassword in the customer table must be specified.
Group:	Framework

SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Customer::AuthModule::DB::CustomerPassword'} =  'pw';`

2.48.6. Customer::AuthModule::DB::DSN

Description:	If "DB" was selected for Customer::AuthModule, the DSN for the connection to the customer table must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Customer::AuthModule::DB::DSN'} =`
	`'DBI:mysql:database=customerdb;host=customerdbhost';`

2.48.7. Customer::AuthModule::DB::User

Description:	If "DB" was selected for Customer::AuthModule, a username to connect to the customer table can be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::DB::User'} =  'some_user';`

2.48.8. Customer::AuthModule::DB::Password

Description:	If "DB" was selected for Customer::AuthModule, a password to connect to the customer table can be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::DB::Password'} =  'some_password';`

2.48.9. Customer::AuthModule::DB::Type

Description:	If "DB" was selected for Customer::AuthModule, a database driver (normally autodetection is used) can be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::DB::Type'} =  'mysql';`

2.48.10. Customer::AuthModule::HTTPBasicAuth::Replace

Description:	If "HTTPBasicAuth" was selected for Customer::AuthModule, you can specify to strip leading parts of user names (e. g. for domains like example_domain\user to user).
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::HTTPBasicAuth::Replace'} = 'example_domain\\\\';`

2.48.11. Customer::AuthModule::HTTPBasicAuth::ReplaceRegExp

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, $1 will be the new Login.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::HTTPBasicAuth::ReplaceRegExp'} =  '^(.+?)@.+?$';`

2.48.12. Customer::AuthModule::LDAP::Host

Description:	If "LDAP" was selected for Customer::AuthModule, the LDAP host can be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::Host'} =  'ldap.example.com';`

2.48.13. Customer::AuthModule::LDAP::BaseDN

Description:	If "LDAP" was selected for Customer::AuthModule, the BaseDN must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::BaseDN'} =  'dc=example,dc=com';`

2.48.14. Customer::AuthModule::LDAP::UID

Description:	If "LDAP" was selected for Customer::AuthModule, the user identifier must be specified.

Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::UID'} =  'uid';`

2.48.15. Customer::AuthModule::LDAP::GroupDN

Description:	If "LDAP" was selected for Customer::Authmodule, you can check if the user is allowed to authenticate because he is in a posixGroup, e.g. user needs to be in a group xyz to use OTRS. Specify the group, who may access the system.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::GroupDN'} =`
	`'cn=otrsallow,ou=posixGroups,dc=example,dc=com';`

2.48.16. Customer::AuthModule::LDAP::AccessAttr

Description:	If "LDAP" was selected for Customer::AuthModule, you can specify access attributes here.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::AccessAttr'} =  'memberUid';`

2.48.17. Customer::AuthModule::LDAP::UserAttr

Description:	If "LDAP" was selected for Customer::AuthModule, user attributes can be specified. For LDAP posixGroups use UID, for non LDAP posixGroups use full user DN.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::UserAttr'} =  'UID';`

2.48.18. Customer::AuthModule::LDAP::SearchUserDN

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:	`$Self->{'Customer::AuthModule::LDAP::SearchUserDN'} = 'cn=binduser,ou=users,dc=example,dc=com';`

2.48.19. Customer::AuthModule::LDAP::SearchUserPw

Description:	If "LDAP" was selected for Customer::AuthModule and your users have only anonymous access to the LDAP tree, but you want to search through the data, you can do this with a user who has access to the LDAP directory. Specify the password for this special user here.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::SearchUserPw'} = 'some_password';`

2.48.20. Customer::AuthModule::LDAP::AlwaysFilter

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:	`$Self->{'Customer::AuthModule::LDAP::AlwaysFilter'} =  '(!objectclass=computer)';`

2.48.21. Customer::AuthModule::LDAP::UserSuffix

Description:	If "LDAP" was selected for Customer::AuthModule and if you want to add a suffix to every customer login name, specifiy 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:	`$Self->{'Customer::AuthModule::LDAP::UserSuffix'} =  '@domain.com';`

2.48.22. Customer::AuthModule::LDAP::Params

Description:	If "LDAP" was selected for Customer::AuthModule and special paramaters are needed for the Net::LDAP perl module, you can specify them here. See "perldoc Net::LDAP" for more information about the parameters.

Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0

Config-Setting:	$Self->{'Customer::AuthModule::LDAP::Params'} = {
	  'async' => '0',
	  'port' => '389',
	  'timeout' => '120',
	  'version' => '3'
	};

2.48.23. Customer::AuthModule::LDAP::Die

Description:	If "LDAP" was selected for Customer::AuthModule, you can specify if the applications will stop if e. g. a connection to a server can't be established due to network problems.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	0
Config-Setting:	$Self->{'Customer::AuthModule::LDAP::Die'} = '1';

2.48.24. Customer::AuthModule::Radius::Host

Description:	If "Radius" was selected for Customer::AuthModule, the radius host must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	$Self->{'Customer::AuthModule::Radius::Host'} = 'radiushost';

2.48.25. Customer::AuthModule::Radius::Password

Description:	If "Radius" was selected for Customer::AuthModule, the password to authenticate to the radius host must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	$Self->{'Customer::AuthModule::Radius::Password'} = 'radiussecret';

2.48.26. Customer::AuthModule::Radius::Die

Description:	If "Radius" was selected for Customer::AuthModule, you can specify if the applications will stop if e. g. a connection to a server can't be established due to network problems.

Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::Radius::Die'} =  '1';`

2.49. Frontend::Customer::ModuleMetaHead

2.49.1. CustomerFrontend::HeaderMetaModule###1-Refresh

Description:	Defines the module to generate html refresh headers of html sites, in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleMetaHead
Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerFrontend::HeaderMetaModule'}->{'1-Refresh'} =  {`
	`  'Module' => 'Kernel::Output::HTML::HeaderMetaRefresh'`
	`};`

2.50. Frontend::Customer::ModuleNotify

2.50.1. CustomerFrontend::NotifyModule###1-ShowAgentOnline

Description:	Defines the module that shows the currently loged in agents in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleNotify
Valid:	0
Required:	0

Config-Setting:	`$Self->{'CustomerFrontend::NotifyModule'}->{'1-ShowAgentOnline'} =  {`
	`  'IdleMinutes' => '60',`
	`  'Module' => 'Kernel::Output::HTML::NotificationAgentOnline',`
	`  'ShowEmail' => '1'`
	`};`

2.50.2. CustomerFrontend::NotifyModule###1-ShowCustomerOnline

Description:	Defines the module that shows the currently loged in customers in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleNotify
Valid:	0
Required:	0

Config-Setting:	`$Self->{'CustomerFrontend::NotifyModule'}->{'1-ShowCustomerOnline'} =`
	`  {`
	`  'Module' => 'Kernel::Output::HTML::NotificationCustomerOnline',`
	`  'ShowEmail' => '1'`
	`};`

2.51. Frontend::Customer::ModuleRegistration

2.51.1. CustomerFrontend::Module###Logout

Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'Logout'} =  {`
	`  'Description' => 'Logout of customer panel',`
	`  'NavBarName' => '',`
	`  'Title' => ''`
	`};`

2.51.2. CustomerFrontend::Module###CustomerPreferences

Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerPreferences'} =  {`
	`  'Description' => 'Customer preferences',`
	`  'NavBarName' => '',`
	`  'Title' => 'Preferences'`
	`};`

2.51.3. CustomerFrontend::Module###CustomerAccept

Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerAccept'} =  {`
	`  'Description' => 'To accept login information, such as an EULA or license.',`
	`  'NavBarName' => '',`
	`  'Title' => 'Info'`
	`};`

2.51.4. CustomerFrontend::Module###PictureUpload

Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'PictureUpload'} =  {`
	`  'Description' => 'Picture upload module',`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'Picture-Upload'`
	`};`

2.52. Frontend::Customer::Preferences

2.52.1. PreferencesTable

Description:	Defines the name of the table, where the customer preferences are stored.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PreferencesTable'} =  'user_preferences';`

2.52.2. PreferencesTableKey

Description:	Defines the column to store the keys for the preferences table.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PreferencesTableKey'} =  'preferences_key';`

2.52.3. CustomerPreferences

Description:	Defines the parameters for the customer preferences table.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerPreferences'} =  {`
	`  'Module' => 'Kernel::System::CustomerUser::Preferences::DB',`
	`  'Params' => {`
	`    'Table' => 'customer_preferences',`
	`    'TableKey' => 'preferences_key',`
	`    'TableUserID' => 'user_id',`
	`    'TableValue' => 'preferences_value'`
	`  }`
	`};`

2.52.4. CustomerPreferencesView

Description:	Sets the order of the different items in the customer preferences view.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences

Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerPreferencesView'} =  [`
	`  'User Profile',`
	`  'Other Settings'`
	`];`

2.52.5. CustomerPreferencesGroups###Password

Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerPreferencesGroups'}->{'Password'} =  {`
	`  'Active' => '1',`
	`  'Area' => 'Customer',`
	`  'Column' => 'Other Settings',`
	`  'Label' => 'Change password',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesPassword',`
	`  'PasswordMin2Characters' => '0',`
	`  'PasswordMin2Lower2UpperCharacters' => '0',`
	`  'PasswordMinSize' => '0',`
	`  'PasswordNeedDigit' => '0',`
	`  'PasswordRegExp' => '',`
	`  'Prio' => '1000'`
	`};`

2.52.6. CustomerPreferencesGroups###Language

Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerPreferencesGroups'}->{'Language'} =  {`
	`  'Active' => '1',`
	`  'Column' => 'User Profile',`
	`  'Key' => 'Language',`
	`  'Label' => 'Interface language',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesLanguage',`
	`  'PrefKey' => 'UserLanguage',`
	`  'Prio' => '2000'`
	`};`

2.52.7. CustomerPreferencesGroups###Theme

Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0

Config-Setting:	$Self->{'CustomerPreferencesGroups'}->{'Theme'} = {
	  'Active' => '0',
	  'Column' => 'User Profile',
	  'Key' => 'Select your frontend Theme.',
	  'Label' => 'Theme',
	  'Module' => 'Kernel::Output::HTML::PreferencesTheme',
	  'PrefKey' => 'UserTheme',
	  'Prio' => '1000'
	};

2.52.8. CustomerPreferencesGroups###TimeZone

Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0

Config-Setting:	$Self->{'CustomerPreferencesGroups'}->{'TimeZone'} = {
	  'Active' => '1',
	  'Column' => 'User Profile',
	  'Key' => 'Time Zone',
	  'Label' => 'Time Zone',
	  'Module' => 'Kernel::Output::HTML::PreferencesTimeZone',
	  'PrefKey' => 'UserTimeZone',
	  'Prio' => '5000'
	};

2.52.9. CustomerPreferencesGroups###PGP

Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0

Config-Setting:	$Self->{'CustomerPreferencesGroups'}->{'PGP'} = {
	  'Active' => '1',
	  'Column' => 'Other Settings',
	  'Key' => 'PGP Key Upload',
	  'Label' => 'PGP Key',
	  'Module' => 'Kernel::Output::HTML::PreferencesPGP',
	  'PrefKey' => 'UserPGPKey',
	  'Prio' => '10000'
	};

2.52.10. CustomerPreferencesGroups###SMIME

Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	$Self->{'CustomerPreferencesGroups'}->{'SMIME'} = { 'Active' => '1',

	`'Column' => 'Other Settings',`
	`'Key' => 'S/MIME Certificate Upload',`
	`'Label' => 'S/MIME Certificate',`
	`'Module' => 'Kernel::Output::HTML::PreferencesSMIME',`
	`'PrefKey' => 'UserSMIMEKey',`
	`'Prio' => '11000'`
	`};`

2.53. Frontend::Public

2.53.1. PublicFrontend::CommonParam###Action

Description:	Defines the default value for the action parameter for the public frontend. The action parameter is used in the scripts of the system.
Group:	Framework
SubGroup:	Frontend::Public
Valid:	1
Required:	1

Config-Setting:	`$Self->{'PublicFrontend::CommonParam'}->{'Action'} =`
	`'PublicDefault';`

2.54. Frontend::Public::ModuleRegistration

2.54.1. PublicFrontend::Module###PublicDefault

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Public::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PublicFrontend::Module'}->{'PublicDefault'} =  {`
	`'Description' => 'PublicDefault',`
	`'NavBarName' => '',`
	`'Title' => 'PublicDefault'`
	`};`

2.54.2. PublicFrontend::Module###PublicRepository

Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Public::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PublicFrontend::Module'}->{'PublicRepository'} =  {`
	`'Description' => 'PublicRepository',`
	`'NavBarName' => '',`
	`'Title' => 'PublicRepository'`
	`};`

2.54.3. PublicFrontend::Module###PublicSupportDataCollector

Description:	Frontend module registration for the agent interface.

Group:	Framework
SubGroup:	Frontend::Public::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'PublicFrontend::Module'}->{'PublicSupportDataCollector'} =
	{
	  'Description' => 'PublicSupportDataCollector',
	  'NavBarName' => '',
	  'Title' => 'PublicSupportDataCollector'
	};

3. GenericInterface

3.1. Core::Ticket

3.1.1. Ticket::EventModulePost###1000-GenericInterface

Description:	Performs the configured action for each event (as an Invoker) for each configured Webservice.
Group:	GenericInterface
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	$Self->{'Ticket::EventModulePost'}->{'1000-GenericInterface'} = {
	  'Event' => '',
	  'Module' => 'Kernel::GenericInterface::Event::Handler',
	  'Transaction' => '1'
	};

3.2. Frontend::Admin::ModuleRegistration

3.2.1. Frontend::Module###AdminGenericInterfaceDebugger

Description:	Frontend module registration for the agent interface.
Group:	GenericInterface
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AdminGenericInterfaceDebugger'} = {
	  'Description' => 'Admin',
	  'Group' => [
	    'admin'
	  ],
	  'Loader' => {
	    'CSS' => [
	      'Core.Agent.Admin.GenericInterface.css'
	    ],
	    'CSS_IE7' => 'Core.Agent.Admin.GenericInterface.IE7.css',
	    'JavaScript' => [
	      'Core.Agent.Admin.GenericInterfaceDebugger.js'
	    ]
	  },
	  'Title' => 'GenericInterface Debugger GUI'
	};

3.2.2. Frontend::Module###AdminGenericInterfaceWebservice

Description:	Frontend module registration for the agent interface.
Group:	GenericInterface
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminGenericInterfaceWebservice'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.GenericInterface.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.GenericInterfaceWebservice.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Create and manage web services.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Web Services',
    'Prio' => '1000'
  },
  'NavBarName' => 'Admin',
  'Title' => 'GenericInterface Web Service GUI'
};
```

3.2.3. Frontend::Module###AdminGenericInterfaceTransportHTTPSOAP

Description:	Frontend module registration for the agent interface.
Group:	GenericInterface
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminGenericInterfaceTransportHTTPSOAP'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.GenericInterface.css'
    ]
  },
  'Title' => 'GenericInterface TransportHTTPSOAP GUI'
};
```

3.2.4. Frontend::Module###AdminGenericInterfaceTransportHTTPREST

Description:	Frontend module registration for the agent interface.
Group:	GenericInterface

SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}-`
	`>{'AdminGenericInterfaceTransportHTTPREST'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'Loader' => {`
	`    'CSS' => [`
	`      'Core.Agent.Admin.GenericInterface.css'`
	`    ]`
	`  },`
	`  'Title' => 'GenericInterface TransportHTTPREST GUI'`
	`};`

3.2.5. Frontend::Module###AdminGenericInterfaceWebserviceHistory

Description:	Frontend module registration for the agent interface.
Group:	GenericInterface
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}-`
	`>{'AdminGenericInterfaceWebserviceHistory'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'Loader' => {`
	`    'CSS' => [`
	`      'Core.Agent.Admin.GenericInterface.css'`
	`    ],`
	`    'JavaScript' => [`
	`      'Core.Agent.Admin.GenericInterfaceWebserviceHistory.js'`
	`    ]`
	`  },`
	`  'Title' => 'GenericInterface Webservice History GUI'`
	`};`

3.2.6. Frontend::Module###AdminGenericInterfaceOperationDefault

Description:	Frontend module registration for the agent interface.
Group:	GenericInterface
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}-`
	`>{'AdminGenericInterfaceOperationDefault'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'Loader' => {`

```
    'CSS' => [
      'Core.Agent.Admin.GenericInterface.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.GenericInterfaceOperation.js'
    ]
  },
  'Title' => 'GenericInterface Operation GUI'
};
```

3.2.7. Frontend::Module###AdminGenericInterfaceInvokerDefault

Description:	Frontend module registration for the agent interface.
Group:	GenericInterface
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
$Self->{'Frontend::Module'}->{'AdminGenericInterfaceInvokerDefault'}
 =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.GenericInterface.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.GenericInterfaceInvoker.js'
    ]
  },
  'Title' => 'GenericInterface Invoker GUI'
};
```

3.2.8. Frontend::Module###AdminGenericInterfaceMappingSimple

Description:	Frontend module registration for the agent interface.
Group:	GenericInterface
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
$Self->{'Frontend::Module'}->{'AdminGenericInterfaceMappingSimple'} =
  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.GenericInterface.css'
    ],
    'CSS_IE7' => 'Core.Agent.Admin.GenericInterface.IE7.css',
    'JavaScript' => [
      'Core.Agent.Admin.GenericInterfaceMappingSimple.js'
    ]
  },
  'Title' => 'GenericInterface Webservice Mapping GUI'
};
```

3.3. GenericInterface::Invoker::ModuleRegistration

3.3.1. GenericInterface::Invoker::Module###Test::Test

Description:	GenericInterface module registration for the invoker layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Invoker::ModuleRegistration
Valid:	0
Required:	0

Config-Setting:	`$Self->{'GenericInterface::Invoker::Module'}->{'Test::Test'} =  {`
	`  'ConfigDialog' => 'AdminGenericInterfaceInvokerDefault',`
	`  'Controller' => 'Test',`
	`  'Name' => 'Test'`
	`};`

3.3.2. GenericInterface::Invoker::Module###Test::TestSimple

Description:	GenericInterface module registration for the invoker layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Invoker::ModuleRegistration
Valid:	0
Required:	0

Config-Setting:	`$Self->{'GenericInterface::Invoker::Module'}->{'Test::TestSimple'} =`
	`{`
	`  'ConfigDialog' => 'AdminGenericInterfaceInvokerDefault',`
	`  'Controller' => 'Test',`
	`  'Name' => 'TestSimple'`
	`};`

3.4. GenericInterface::Mapping::ModuleRegistration

3.4.1. GenericInterface::Mapping::Module###Test

Description:	GenericInterface module registration for the mapping layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Mapping::ModuleRegistration
Valid:	0
Required:	0

Config-Setting:	`$Self->{'GenericInterface::Mapping::Module'}->{'Test'} =  {`
	`  'ConfigDialog' => ''`
	`};`

3.4.2. GenericInterface::Mapping::Module###Simple

Description:	GenericInterface module registration for the mapping layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Mapping::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'GenericInterface::Mapping::Module'}->{'Simple'} =  {`
	`  'ConfigDialog' => 'AdminGenericInterfaceMappingSimple'`

	};

3.5. GenericInterface::Operation::ModuleRegistration

3.5.1. GenericInterface::Operation::Module###Test::Test

Description:	GenericInterface module registration for the operation layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::ModuleRegistration
Valid:	0
Required:	0

Config-Setting:	$Self->{'GenericInterface::Operation::Module'}->{'Test::Test'} = {
	'ConfigDialog' => 'AdminGenericInterfaceOperationDefault',
	'Controller' => 'Test',
	'Name' => 'Test'
	};

3.5.2. GenericInterface::Operation::Module###Session::SessionCreate

Description:	GenericInterface module registration for the operation layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'GenericInterface::Operation::Module'}-
	>{'Session::SessionCreate'} = {
	'ConfigDialog' => 'AdminGenericInterfaceOperationDefault',
	'Controller' => 'Session',
	'Name' => 'SessionCreate'
	};

3.5.3. GenericInterface::Operation::Module###Ticket::TicketCreate

Description:	GenericInterface module registration for the operation layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'GenericInterface::Operation::Module'}-
	>{'Ticket::TicketCreate'} = {
	'ConfigDialog' => 'AdminGenericInterfaceOperationDefault',
	'Controller' => 'Ticket',
	'Name' => 'TicketCreate'
	};

3.5.4. GenericInterface::Operation::Module###Ticket::TicketUpdate

Description:	GenericInterface module registration for the operation layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::ModuleRegistration
Valid:	1

Required:	0

Config-Setting:	$Self->{'GenericInterface::Operation::Module'}-
	>{'Ticket::TicketUpdate'} = {
	  'ConfigDialog' => 'AdminGenericInterfaceOperationDefault',
	  'Controller' => 'Ticket',
	  'Name' => 'TicketUpdate'
	};

3.5.5. GenericInterface::Operation::Module###Ticket::TicketGet

Description:	GenericInterface module registration for the operation layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'GenericInterface::Operation::Module'}->{'Ticket::TicketGet'}
	 = {
	  'ConfigDialog' => 'AdminGenericInterfaceOperationDefault',
	  'Controller' => 'Ticket',
	  'Name' => 'TicketGet'
	};

3.5.6. GenericInterface::Operation::Module###Ticket::TicketSearch

Description:	GenericInterface module registration for the operation layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'GenericInterface::Operation::Module'}-
	>{'Ticket::TicketSearch'} = {
	  'ConfigDialog' => 'AdminGenericInterfaceOperationDefault',
	  'Controller' => 'Ticket',
	  'Name' => 'TicketGet'
	};

3.6. GenericInterface::Operation::TicketCreate

3.6.1. GenericInterface::Operation::TicketCreate###ArticleType

Description:	Defines the default type of the article for this operation.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketCreate
Valid:	1
Required:	1
Config-Setting:	$Self->{'GenericInterface::Operation::TicketCreate'}->{'ArticleType'} = 'webrequest';

3.6.2. GenericInterface::Operation::TicketCreate###HistoryType

Description:	Defines the history type for this operation, which gets used for ticket history in the agent interface.

Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketCreate
Valid:	1
Required:	1
Config-Setting:	$Self->{'GenericInterface::Operation::TicketCreate'}->{'HistoryType'} = 'NewTicket';

3.6.3. GenericInterface::Operation::TicketCreate###HistoryComment

Description:	Defines the history comment for this operation, which gets used for ticket history in the agent interface.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketCreate
Valid:	1
Required:	1
Config-Setting:	$Self->{'GenericInterface::Operation::TicketCreate'}- >{'HistoryComment'} = '%%GenericInterface Create';

3.6.4. GenericInterface::Operation::TicketCreate###AutoResponseType

Description:	Defines the default auto response type of the article for this operation.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketCreate
Valid:	1
Required:	1
Config-Setting:	$Self->{'GenericInterface::Operation::TicketCreate'}- >{'AutoResponseType'} = 'auto reply';

3.7. GenericInterface::Operation::TicketSearch

3.7.1. GenericInterface::Operation::TicketSearch###SearchLimit

Description:	Maximum number of tickets to be displayed in the result of this operation.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketSearch
Valid:	1
Required:	1
Config-Setting:	$Self->{'GenericInterface::Operation::TicketSearch'}->{'SearchLimit'} = '500';

3.7.2. GenericInterface::Operation::TicketSearch###SortBy::Default

Description:	Defines the default ticket attribute for ticket sorting of the ticket search result of this operation.
Group:	GenericInterface

SubGroup:	GenericInterface::Operation::TicketSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketSearch'}->{'SortBy::Default'} =  'Age';`

3.7.3. GenericInterface::Operation::TicketSearch###Order::Default

Description:	Defines the default ticket order in the ticket search result of the this operation. Up: oldest on top. Down: latest on top.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketSearch'}->{'Order::Default'} =  'Down';`

3.8. GenericInterface::Operation::TicketUpdate

3.8.1. GenericInterface::Operation::TicketUpdate###ArticleType

Description:	Defines the default type of the article for this operation.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketUpdate
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketUpdate'}->{'ArticleType'} =  'webrequest';`

3.8.2. GenericInterface::Operation::TicketUpdate###HistoryType

Description:	Defines the history type for this operation, which gets used for ticket history in the agent interface.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketUpdate
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketUpdate'}->{'HistoryType'} =  'AddNote';`

3.8.3. GenericInterface::Operation::TicketUpdate###HistoryComment

Description:	Defines the history comment for this operation, which gets used for ticket history in the agent interface.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketUpdate
Valid:	1

Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketUpdate'}->{'HistoryComment'} =  '%%GenericInterface Note';`

3.8.4. GenericInterface::Operation::TicketUpdate###AutoResponseType

Description:	Defines the default auto response type of the article for this operation.
Group:	GenericInterface
SubGroup:	GenericInterface::Operation::TicketUpdate
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::TicketUpdate'}->{'AutoResponseType'} =  'auto follow up';`

3.9. GenericInterface::Transport::ModuleRegistration

3.9.1. GenericInterface::Transport::Module###HTTP::SOAP

Description:	GenericInterface module registration for the transport layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Transport::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'GenericInterface::Transport::Module'}->{'HTTP::SOAP'} =  {`
	`  'ConfigDialog' => 'AdminGenericInterfaceTransportHTTPSOAP',`
	`  'Name' => 'SOAP',`
	`  'Protocol' => 'HTTP'`
	`};`

3.9.2. GenericInterface::Transport::Module###HTTP::REST

Description:	GenericInterface module registration for the transport layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Transport::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'GenericInterface::Transport::Module'}->{'HTTP::REST'} =  {`
	`  'ConfigDialog' => 'AdminGenericInterfaceTransportHTTPREST',`
	`  'Name' => 'REST',`
	`  'Protocol' => 'HTTP'`
	`};`

3.9.3. GenericInterface::Transport::Module###HTTP::Test

Description:	GenericInterface module registration for the transport layer.
Group:	GenericInterface
SubGroup:	GenericInterface::Transport::ModuleRegistration
Valid:	0

Required:	0

Config-Setting:	`$Self->{'GenericInterface::Transport::Module'}->{'HTTP::Test'} =  {`
	`  'ConfigDialog' => 'AdminGenericInterfaceTransportHTTPTest',`
	`  'Name' => 'Test',`
	`  'Protocol' => 'HTTP'`
	`};`

3.10. GenericInterface::Webservice

3.10.1. GenericInterface::WebserviceConfig::CacheTTL

Description:	Cache time in seconds for the web service config backend.
Group:	GenericInterface
SubGroup:	GenericInterface::Webservice
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::WebserviceConfig::CacheTTL'} =  '86400';`

3.10.2. GenericInterface::Operation::Common::CachedAuth::AgentCacheTTL

Description:	Cache time in seconds for agent authentication in the GenericInterface.
Group:	GenericInterface
SubGroup:	GenericInterface::Webservice
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::Common::CachedAuth::AgentCacheTTL'} =  '300';`

3.10.3. GenericInterface::Operation::Common::CachedAuth::CustomerCacheTTL

Description:	Cache time in seconds for customer authentication in the GenericInterface.
Group:	GenericInterface
SubGroup:	GenericInterface::Webservice
Valid:	1
Required:	1
Config-Setting:	`$Self->{'GenericInterface::Operation::Common::CachedAuth::CustomerCacheTTL'} =  '300';`

3.10.4. GenericInterface::Webservice::Path::Separator

Description:	Webservice path separator.
Group:	GenericInterface
SubGroup:	GenericInterface::Webservice
Valid:	1

Required:	1
Config-Setting:	$Self->{'GenericInterface::Webservice::Path::Separator'} = '»';

4. ProcessManagement

4.1. Core

4.1.1. Process::DynamicFieldProcessManagementProcessID

Description:	This option defines the dynamic field in which a Process Management process entity id is stored.
Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	$Self->{'Process::DynamicFieldProcessManagementProcessID'} = 'ProcessManagementProcessID';

4.1.2. Process::DynamicFieldProcessManagementActivityID

Description:	This option defines the dynamic field in which a Process Management activity entity id is stored.
Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	$Self->{'Process::DynamicFieldProcessManagementActivityID'} = 'ProcessManagementActivityID';

4.1.3. Process::DefaultQueue

Description:	This option defines the process tickets default queue.
Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	$Self->{'Process::DefaultQueue'} = 'Raw';

4.1.4. Process::DefaultState

Description:	This option defines the process tickets default state.
Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	$Self->{'Process::DefaultState'} = 'new';

4.1.5. Process::DefaultLock

Description:	This option defines the process tickets default lock.
Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Process::DefaultLock'} =  'unlock';`

4.1.6. Process::DefaultPriority

Description:	This option defines the process tickets default priority.
Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Process::DefaultPriority'} =  '3 normal';`

4.1.7. Process::Entity::Prefix

Description:	Default ProcessManagement entity prefixes for entity IDs that are automatically generated.
Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Process::Entity::Prefix'} =  {`
	`  'Activity' => 'A',`
	`  'ActivityDialog' => 'AD',`
	`  'Process' => 'P',`
	`  'Transition' => 'T',`
	`  'TransitionAction' => 'TA'`
	`};`

4.1.8. Process::CacheTTL

Description:	Cache time in seconds for the DB process backend.
Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Process::CacheTTL'} =  '3600';`

4.1.9. Process::NavBarOutput::CacheTTL

Description:	Cache time in seconds for the ticket process navigation bar output module.

Group:	ProcessManagement
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Process::NavBarOutput::CacheTTL'} =  '900';`

4.2. Core::Ticket

4.2.1. Ticket::EventModulePost###TicketProcessTransitions

Description:	Event module registration. For more performance you can define a trigger event (e. g. Event => TicketCreate).
Group:	ProcessManagement
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'TicketProcessTransitions'} =  {`
	`  'Event' => '',`
	`  'Module' =>`
	`'Kernel::System::Ticket::Event::TicketProcessTransitions',`
	`  'Transaction' => '1'`
	`};`

4.3. Core::Transition

4.3.1. ProcessManagement::Transition::Debug::Enabled

Description:	If enabled debugging information for transitions is logged.
Group:	ProcessManagement
SubGroup:	Core::Transition
Valid:	1
Required:	1
Config-Setting:	`$Self->{'ProcessManagement::Transition::Debug::Enabled'} =  '0';`

4.3.2. ProcessManagement::Transition::Debug::LogPriority

Description:	Defines the priority in which the information is logged and presented.
Group:	ProcessManagement
SubGroup:	Core::Transition
Valid:	0
Required:	0

Config-Setting:	`$Self->{'ProcessManagement::Transition::Debug::LogPriority'} =`
	`'debug';`

4.3.3. ProcessManagement::Transition::Debug::Filter###00-Default

Description:	Filter for debugging Transitions. Note: More filters can be added in the format <OTRS_TICKET_Attribute> e.g. <OTRS_TICKET_Priority>.

Group:	ProcessManagement
SubGroup:	Core::Transition
Valid:	0
Required:	0

Config-Setting:	$Self->{'ProcessManagement::Transition::Debug::Filter'}->{'00-Default'} = {
	  '<OTRS_TICKET_TicketNumber>' => '',
	  'TransitionEntityID' => ''
	};

4.4. DynamicFields::Driver::Registration

4.4.1. DynamicFields::Driver###ProcessID

Description:	DynamicField backend registration.
Group:	ProcessManagement
SubGroup:	DynamicFields::Driver::Registration
Valid:	1
Required:	0

Config-Setting:	$Self->{'DynamicFields::Driver'}->{'ProcessID'} = {
	  'ConfigDialog' => 'AdminDynamicFieldText',
	  'DisabledAdd' => '1',
	  'DisplayName' => 'ProcessID',
	  'Module' =>
	 'Kernel::System::DynamicField::Driver::ProcessManagement::ProcessID'
	};

4.4.2. DynamicFields::Driver###ActivityID

Description:	DynamicField backend registration.
Group:	ProcessManagement
SubGroup:	DynamicFields::Driver::Registration
Valid:	1
Required:	0

Config-Setting:	$Self->{'DynamicFields::Driver'}->{'ActivityID'} = {
	  'ConfigDialog' => 'AdminDynamicFieldText',
	  'DisabledAdd' => '1',
	  'DisplayName' => 'ActivityID',
	  'Module' =>
	 'Kernel::System::DynamicField::Driver::ProcessManagement::ActivityID'
	};

4.5. Frontend::Admin::ModuleRegistration

4.5.1. Frontend::Module###AdminProcessManagement

Description:	Frontend module registration for the agent interface.
Group:	ProcessManagement
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
$Self->{'Frontend::Module'}->{'AdminProcessManagement'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ProcessManagement.css',
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'thirdparty/jsplumb-1.6.1/jsplumb.js',
      'thirdparty/farahey-0.5/farahey.js',
      'thirdparty/jsplumb-labelspacer/label-spacer.js',
      'Core.Agent.Admin.ProcessManagement.js',
      'Core.Agent.Admin.ProcessManagement.Canvas.js',
      'Core.UI.AllocationList.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Configure Processes.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Process Management',
    'Prio' => '750'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Process Management'
};
```

4.5.2. Frontend::Module###AdminProcessManagementActivity

Description:	Frontend module registration for the agent interface.
Group:	ProcessManagement
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
$Self->{'Frontend::Module'}->{'AdminProcessManagementActivity'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ProcessManagement.css',
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ProcessManagement.js',
      'Core.UI.AllocationList.js'
    ]
  },
  'Title' => 'Process Management Activity GUI'
};
```

4.5.3. Frontend::Module###AdminProcessManagementActivityDialog

Description:	Frontend module registration for the agent interface.
Group:	ProcessManagement
SubGroup:	Frontend::Admin::ModuleRegistration

Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminProcessManagementActivityDialog'}
 =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ProcessManagement.css',
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ProcessManagement.js',
      'Core.UI.AllocationList.js'
    ]
  },
  'Title' => 'Process Management Activity Dialog GUI'
};
```

4.5.4. Frontend::Module###AdminProcessManagementTransition

Description:	Frontend module registration for the agent interface.
Group:	ProcessManagement
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminProcessManagementTransition'} =
 {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ProcessManagement.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ProcessManagement.js'
    ]
  },
  'Title' => 'Process Management Transition GUI'
};
```

4.5.5. Frontend::Module###AdminProcessManagementTransitionAction

Description:	Frontend module registration for the agent interface.
Group:	ProcessManagement
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}-
>{'AdminProcessManagementTransitionAction'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ProcessManagement.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ProcessManagement.js'
    ]
  },
  'Title' => 'Process Management Transition Action GUI'
};
```

4.5.6. Frontend::Module###AdminProcessManagementPath

Description:	Frontend module registration for the agent interface.
Group:	ProcessManagement
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AdminProcessManagementPath'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ProcessManagement.css',
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ProcessManagement.js',
      'Core.UI.AllocationList.js'
    ]
  },
  'Title' => 'Process Management Path GUI'
};
```

4.6. Frontend::Agent::Dashboard

4.6.1. DashboardBackend###0140-RunningTicketProcess

Description:	Parameters for the dashboard backend of the running process tickets overview of the agent interface . "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	ProcessManagement
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'DashboardBackend'}->{'0140-RunningTicketProcess'} =  {
  'Attributes' => 'StateType=new;StateType=open;StateType=pending
reminder;StateType=pending auto',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '0',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'DynamicField_ProcessManagementActivityID' => '2',
    'DynamicField_ProcessManagementProcessID' => '2',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All tickets with a reminder set where the reminder
 date has been reached',
  'Group' => '',
  'IsProcessWidget' => '1',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
  'Permission' => 'rw',
  'Time' => 'UntilTime',
  'Title' => 'Running Process Tickets'
};
```

4.7. Frontend::Agent::ModuleRegistration

4.7.1. Frontend::Module###AgentTicketProcess

Description:	Frontend module registration for the agent interface.
Group:	ProcessManagement
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AgentTicketProcess'} =  {
  'Description' => 'Create new process ticket',
  'Loader' => {
    'CSS' => [
      'Core.Agent.TicketProcess.css'
    ],
    'JavaScript' => [
      'Core.Agent.CustomerSearch.js',
      'Core.Agent.TicketAction.js',
      'Core.Agent.TicketProcess.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'p',
      'Block' => '',
      'Description' => 'Create New process ticket',
```

	`      'Link' => 'Action=AgentTicketProcess',`
	`      'LinkOption' => '',`
	`      'Name' => 'New process ticket',`
	`      'NavBar' => 'Ticket',`
	`      'Prio' => '220',`
	`      'Type' => ''`
	`    }`
	`  ],`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'New process ticket'`
	`};`

4.8. Frontend::Agent::NavBarModule

4.8.1. Frontend::NavBarModule###1-TicketProcesses

Description:	Frontend module registration (disable ticket processes screen if no process available).
Group:	ProcessManagement
SubGroup:	Frontend::Agent::NavBarModule
Valid:	1
Required:	0

Config-Setting:	` $Self->{'Frontend::NavBarModule'}->{'1-TicketProcesses'} =  {`
	`  'Module' => 'Kernel::Output::HTML::NavBarAgentTicketProcess'`
	`};`

4.9. Frontend::Agent::Ticket::MenuModule

4.9.1. Ticket::Frontend::MenuModule###480-Process

Description:	Shows a link in the menu to enroll a ticket into a process in the ticket zoom view of the agent interface.
Group:	ProcessManagement
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	` $Self->{'Ticket::Frontend::MenuModule'}->{'480-Process'} =  {`
	`  'Action' => 'AgentTicketProcess',`
	`  'Description' => 'Enroll this ticket into a process',`
	`  'Link' => 'Action=AgentTicketProcess;IsProcessEnroll=1;TicketID=[%`
	` Data.TicketID | html %]',`
	`  'Module' => 'Kernel::Output::HTML::TicketMenuProcess',`
	`  'Name' => 'Process Enroll',`
	`  'PopupType' => 'TicketAction',`
	`  'Target' => ''`
	`};`

4.10. Frontend::Agent::Ticket::ViewProcess

4.10.1. Ticket::Frontend::AgentTicketProcess###StateType

Description:	Determines the next possible ticket states, for process tickets in the agent interface.
Group:	ProcessManagement
SubGroup:	Frontend::Agent::Ticket::ViewProcess

Valid:	1
Required:	1

Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketProcess'}->{'StateType'} = [
	  'new',
	  'open',
	  'pending auto',
	  'pending reminder',
	  'closed'
	];

4.10.2. Ticket::Frontend::CustomerTicketProcess###StateType

Description:	Determines the next possible ticket states, for process tickets in the agent interface.
Group:	ProcessManagement
SubGroup:	Frontend::Agent::Ticket::ViewProcess
Valid:	1
Required:	1

Config-Setting:	$Self->{'Ticket::Frontend::CustomerTicketProcess'}->{'StateType'} = [
	  'new',
	  'open'
	];

4.11. Frontend::Agent::Ticket::ViewZoom

4.11.1. Ticket::Frontend::AgentTicketZoom###ProcessDisplay

Description:	Display settings to override defaults for Process Tickets.
Group:	ProcessManagement
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1

Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketZoom'}->{'ProcessDisplay'} = {
	  'NavBarName' => 'Processes',
	  'WidgetTitle' => 'Process Information'
	};

4.11.2. Ticket::Frontend::AgentTicketZoom###ProcessWidgetDynamicFieldGroups

Description:	Dynamic fields groups for process widget. The key is the name of the group, the value contains the fields to be shown. Example: 'Key => My Group', 'Content: Name_X, NameY'.
Group:	ProcessManagement
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketZoom'}->{'ProcessWidgetDynamicFieldGroups'} = {};

4.11.3. Ticket::Frontend::AgentTicketZoom###ProcessWidgetDynamicField

Description:	Dynamic fields shown in the process widget in ticket zoom screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	ProcessManagement
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketZoom'}->{'ProcessWidgetDynamicField'} =  {};`

4.12. Frontend::Customer::ModuleRegistration

4.12.1. CustomerFrontend::Module###CustomerTicketProcess

Description:	Frontend module registration for the customer interface.
Group:	ProcessManagement
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	see below

```
 $Self->{'CustomerFrontend::Module'}->{'CustomerTicketProcess'} =  {
  'Description' => 'Process Ticket',
  'Loader' => {
    'CSS' => [
      'Core.Customer.TicketProcess.css'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'o',
      'Block' => '',
      'Description' => 'Create new process ticket',
      'Link' => 'Action=CustomerTicketProcess',
      'LinkOption' => '',
      'Name' => 'New process ticket',
      'NavBar' => 'Ticket',
      'Prio' => '220',
      'Type' => 'Submenu'
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'Process ticket'
};
```

4.13. Frontend::Customer::NavBarModule

4.13.1. CustomerFrontend::NavBarModule###10-CustomerTicketProcesses

Description:	Frontend module registration (disable ticket processes screen if no process available) for Customer.
Group:	ProcessManagement
SubGroup:	Frontend::Customer::NavBarModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerFrontend::NavBarModule'}->{'10-CustomerTicketProcesses'} = { 'Module' => 'Kernel::Output::HTML::NavBarCustomerTicketProcess' };`

5. Scheduler

5.1. Core

5.1.1. Scheduler::SleepTime

Description:	Defines scheduler sleep time in seconds after processing all available tasks (floating point number).
Group:	Scheduler
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Scheduler::SleepTime'} =  '1.0';`

5.1.2. Scheduler::PIDUpdateTime

Description:	Defines scheduler PID update time in seconds.
Group:	Scheduler
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Scheduler::PIDUpdateTime'} =  '600';`

5.1.3. Scheduler::RestartAfterSeconds

Description:	Defines the time in seconds after which the Scheduler performs an automatic self-restart.
Group:	Scheduler
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Scheduler::RestartAfterSeconds'} =  '86400';`

5.1.4. Scheduler::TaskDataLength

Description:	Defines the maximum length (in characters) for a scheduler task data. WARNING: Do not modify this setting unless you are sure of the current Database length for 'task_data' filed from 'scheduler_data_list' table.
Group:	Scheduler
SubGroup:	Core
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Scheduler::TaskDataLength'} =  '8000';`

5.2. Core::Log

5.2.1. Scheduler::LogPath

Description:	Defines the path for scheduler to store its console output (SchedulerOUT.log and SchedulerERR.log).
Group:	Scheduler
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Scheduler::LogPath'} =  '<OTRS_CONFIG_Home>/var/log';`

5.2.2. Scheduler::Log::DaysToKeep

Description:	Defines the time in days to keep log backup files.
Group:	Scheduler
SubGroup:	Core::Log
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Scheduler::Log::DaysToKeep'} =  '10';`

5.3. Core::Web

5.3.1. Loader::Agent::CommonCSS###000-Scheduler

Description:	List of CSS files to always be loaded for the agent interface.
Group:	Scheduler
SubGroup:	Core::Web
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Loader::Agent::CommonCSS'}->{'000-Scheduler'} =  [`
	`  'Core.Agent.SchedulerInfo.css'`
	`];`

5.3.2. Loader::Agent::CommonJS###000-Scheduler

Description:	List of JS files to always be loaded for the agent interface.
Group:	Scheduler
SubGroup:	Core::Web
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Loader::Agent::CommonJS'}->{'000-Scheduler'} =  [`
	`  'Core.Agent.SchedulerInfo.js'`
	`];`

5.4. Frontend::Admin::ModuleRegistration

5.4.1. Frontend::Module###AgentSchedulerInfo

Description:	Frontend module registration for the agent interface.
Group:	Scheduler
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentSchedulerInfo'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'Title' => 'Shows information on how to start OTRS Scheduler'`
	`};`

5.5. Frontend::Agent::ModuleNotify

5.5.1. Frontend::NotifyModule###800-Scheduler-Check

Description:	Defines the module to display a notification in the agent interface if the scheduler is not running.
Group:	Scheduler
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'800-Scheduler-Check'} =  {`
	`  'Module' => 'Kernel::Output::HTML::NotificationSchedulerCheck'`
	`};`

6. Ticket

6.1. Core

6.1.1. OTRSEscalationEvents::DecayTime

Description:	The duration in minutes after emitting an event, in which the new escalation notify and start events are suppressed.
Group:	Ticket
SubGroup:	Core
Valid:	1
Required:	0
Config-Setting:	`$Self->{'OTRSEscalationEvents::DecayTime'} =  '1440';`

6.2. Core::CustomerCompany

6.2.1. CustomerCompany::EventModulePost###110-UpdateTickets

Description:	Event module that updates tickets after an update of the Customer.
Group:	Ticket

SubGroup:	Core::CustomerCompany
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerCompany::EventModulePost'}->{'110-UpdateTickets'} =`
	`{`
	`  'Event' => 'CustomerCompanyUpdate',`
	`  'Module' => 'Kernel::System::CustomerCompany::Event::TicketUpdate',`
	`  'Transaction' => '0'`
	`};`

6.2.2. CustomerUser::EventModulePost###120-UpdateTickets

Description:	Event module that updates tickets after an update of the Customer User.
Group:	Ticket
SubGroup:	Core::CustomerCompany
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerUser::EventModulePost'}->{'120-UpdateTickets'} =  {`
	`  'Event' => 'CustomerUserUpdate',`
	`  'Module' => 'Kernel::System::CustomerUser::Event::TicketUpdate',`
	`  'Transaction' => '0'`
	`};`

6.3. Core::FulltextSearch

6.3.1. Ticket::SearchIndexModule

Description:	Helps to extend your articles full-text search (From, To, Cc, Subject and Body search). Runtime will do full-text searches on live data (it works fine for up to 50.000 tickets). StaticDB will strip all articles and will build an index after article creation, increasing fulltext searches about 50%. To create an initial index use "bin/otrs.RebuildFulltextIndex.pl".
Group:	Ticket
SubGroup:	Core::FulltextSearch
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::SearchIndexModule'} =`
	`'Kernel::System::Ticket::ArticleSearchIndex::RuntimeDB';`

6.3.2. Ticket::SearchIndex::Attribute

Description:	Basic fulltext index settings. Execute "bin/otrs.RebuildFulltextIndex.pl" in order to generate a new index.
Group:	Ticket
SubGroup:	Core::FulltextSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::SearchIndex::Attribute'} =  {`

```
  'WordCountMax' => '1000',
  'WordLengthMax' => '30',
  'WordLengthMin' => '3'
};
```

6.3.3. Ticket::SearchIndex::Filters

Description:	Fulltext index regex filters to remove parts of the text.
Group:	Ticket
SubGroup:	Core::FulltextSearch
Valid:	1
Required:	1

Config-Setting:

```
 $Self->{'Ticket::SearchIndex::Filters'} =  [
  '[,\\&\\<\\>\\?"\\!\\*\\|;\\[\\]\\(\\)\\+\\$\\^=]',
  '^[\':.]|[\':.]$'
];
```

6.3.4. Ticket::SearchIndex::StopWords

Description:	Stop words for fulltext index. These words will be removed.
Group:	Ticket
SubGroup:	Core::FulltextSearch
Valid:	1
Required:	1

Config-Setting:

```
 $Self->{'Ticket::SearchIndex::StopWords'} =  {
  'a' => '1',
  'als' => '1',
  'and' => '1',
  'auf' => '1',
  'das' => '1',
  'den' => '1',
  'der' => '1',
  'die' => '1',
  'for' => '1',
  'im' => '1',
  'in' => '1',
  'is' => '1',
  'of' => '1',
  'the' => '1',
  'to' => '1',
  'und' => '1',
  'vom' => '1',
  'zu' => '1'
};
```

6.3.5. Ticket::EventModulePost###98-ArticleSearchIndex

Description:	Builds an article index right after the article's creation.
Group:	Ticket
SubGroup:	Core::FulltextSearch
Valid:	1
Required:	0

Config-Setting:

```
 $Self->{'Ticket::EventModulePost'}->{'98-ArticleSearchIndex'} =  {
  'Event' => '(ArticleCreate|ArticleUpdate)',
```

	'Module' => 'Kernel::System::Ticket::Event::ArticleSearchIndex' };

6.4. Core::LinkObject

6.4.1. LinkObject::PossibleLink###0200

Description:	Links 2 tickets with a "Normal" type link.
Group:	Ticket
SubGroup:	Core::LinkObject
Valid:	1
Required:	0

Config-Setting:	$Self->{'LinkObject::PossibleLink'}->{'0200'} = {
	'Object1' => 'Ticket',
	'Object2' => 'Ticket',
	'Type' => 'Normal'
	};

6.4.2. LinkObject::PossibleLink###0201

Description:	Links 2 tickets with a "ParentChild" type link.
Group:	Ticket
SubGroup:	Core::LinkObject
Valid:	1
Required:	0

Config-Setting:	$Self->{'LinkObject::PossibleLink'}->{'0201'} = {
	'Object1' => 'Ticket',
	'Object2' => 'Ticket',
	'Type' => 'ParentChild'
	};

6.4.3. LinkObject::IgnoreLinkedTicketStateTypes

Description:	Defines, which tickets of which ticket state types should not be listed in linked ticket lists.
Group:	Ticket
SubGroup:	Core::LinkObject
Valid:	1
Required:	0
Config-Setting:	$Self->{'LinkObject::IgnoreLinkedTicketStateTypes'} = ['merged', 'removed'];

6.5. Core::PostMaster

6.5.1. PostmasterMaxEmails

Description:	Maximal auto email responses to own email-address a day (Loop-Protection).
Group:	Ticket

SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterMaxEmails'} =  '40';`

6.5.2. PostMasterMaxEmailSize

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:	`$Self->{'PostMasterMaxEmailSize'} =  '16384';`

6.5.3. PostMasterReconnectMessage

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:	`$Self->{'PostMasterReconnectMessage'} =  '20';`

6.5.4. LoopProtectionModule

Description:	Default loop protection module.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1

Config-Setting:	`$Self->{'LoopProtectionModule'} =`
	`'Kernel::System::PostMaster::LoopProtection::DB';`

6.5.5. LoopProtectionLog

Description:	Path for the log file (it only applies if "FS" was selected for LoopProtectionModule and it is mandatory).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1

Config-Setting:	`$Self->{'LoopProtectionLog'} =  '<OTRS_CONFIG_Home>/var/log/`
	`LoopProtection';`

6.5.6. PostmasterAutoHTML2Text

Description:	Converts HTML mails into text messages.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterAutoHTML2Text'} =  '1';`

6.5.7. PostmasterFollowUpSearchInReferences

Description:	Executes follow up checks on In-Reply-To or References headers for mails that don't have a ticket number in the subject.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpSearchInReferences'} =  '0';`

6.5.8. PostmasterFollowUpSearchInBody

Description:	Executes follow up mail body checks in mails that don't have a ticket number in the subject.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpSearchInBody'} =  '0';`

6.5.9. PostmasterFollowUpSearchInAttachment

Description:	Executes follow up mail attachments checks in mails that don't have a ticket number in the subject.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpSearchInAttachment'} =  '0';`

6.5.10. PostmasterFollowUpSearchInRaw

Description:	Executes follow up plain/raw mail checks in mails that don't have a ticket number in the subject.
Group:	Ticket
SubGroup:	Core::PostMaster

Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpSearchInRaw'} =  '0';`

6.5.11. PostmasterUserID

Description:	Specifies user id of the postmaster data base.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterUserID'} =  '1';`

6.5.12. PostmasterDefaultQueue

Description:	Defines the postmaster default queue.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterDefaultQueue'} =  'Raw';`

6.5.13. PostmasterDefaultPriority

Description:	Defines the default priority of new tickets.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterDefaultPriority'} =  '3 normal';`

6.5.14. PostmasterDefaultState

Description:	Defines the default state of new tickets.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterDefaultState'} =  'new';`

6.5.15. PostmasterFollowUpState

Description:	Defines the state of a ticket if it gets a follow-up.
Group:	Ticket

SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpState'} =  'open';`

6.5.16. PostmasterFollowUpStateClosed

Description:	Defines the state of a ticket if it gets a follow-up and the ticket was already closed.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	`$Self->{'PostmasterFollowUpStateClosed'} =  'open';`

6.5.17. PostmasterFollowUpOnUnlockAgentNotifyOnlyToOwner

Description:	Sends agent follow-up notification only to the owner, if a ticket is unlocked (the default is to send the notification to all agents).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpOnUnlockAgentNotifyOnlyToOwner'} =  '0';`

6.5.18. PostmasterHeaderFieldCount

Description:	Defines the number of header fields in frontend modules for add and update postmaster filters. It can be up to 99 fields.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterHeaderFieldCount'} =  '12';`

6.5.19. PostmasterX-Header

Description:	Defines all the X-headers that should be scanned.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1

Config-Setting:	`$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',
  'Auto-Submitted',
  'X-Loop',
  'X-Spam-Flag',
  'X-Spam-Level',
  'X-Spam-Score',
  'X-Spam-Status',
  'X-No-Loop',
  'X-Priority',
  'Importance',
  'X-Mailer',
  'User-Agent',
  'Organization',
  'X-Original-To',
  'Delivered-To',
  'Envelope-To',
  'Return-Path',
  'X-OTRS-Owner',
  'X-OTRS-OwnerID',
  'X-OTRS-Responsible',
  'X-OTRS-ResponsibleID',
  'X-OTRS-Loop',
  'X-OTRS-Priority',
  'X-OTRS-Queue',
  'X-OTRS-Lock',
  'X-OTRS-Ignore',
  'X-OTRS-State',
  'X-OTRS-State-PendingTime',
  'X-OTRS-Type',
  'X-OTRS-Service',
  'X-OTRS-SLA',
  'X-OTRS-CustomerNo',
  'X-OTRS-CustomerUser',
  'X-OTRS-SenderType',
  'X-OTRS-ArticleType',
  'X-OTRS-FollowUp-Priority',
  'X-OTRS-FollowUp-Queue',
  'X-OTRS-FollowUp-Lock',
  'X-OTRS-FollowUp-State',
  'X-OTRS-FollowUp-State-PendingTime',
  'X-OTRS-FollowUp-Type',
  'X-OTRS-FollowUp-Service',
  'X-OTRS-FollowUp-SLA',
  'X-OTRS-FollowUp-SenderType',
  'X-OTRS-FollowUp-ArticleType'
];
```

6.5.20. PostMaster::PreFilterModule###1-Match

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:	`$Self->{'PostMaster::PreFilterModule'}->{'1-Match'} =  {`
	`  'Match' => {`
	`    'From' => 'noreply@'`
	`  },`
	`  'Module' => 'Kernel::System::PostMaster::Filter::Match',`
	`  'Set' => {`
	`    'X-OTRS-Ignore' => 'yes'`
	`  },`
	`  'StopAfterMatch' => '0'`
	`};`

6.5.21. PostMaster::PreFilterModule###2-Match

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:	`$Self->{'PostMaster::PreFilterModule'}->{'2-Match'} =  {`
	`  'Match' => {`
	`    'Subject' => 'SomeNumber:(\\d\\d\\d\\d)'`
	`  },`
	`  'Module' => 'Kernel::System::PostMaster::Filter::Match',`
	`  'Set' => {`
	`    'X-OTRS-DynamicField-TicketFreeKey1' => 'SomeNumber',`
	`    'X-OTRS-DynamicField-TicketFreeText1' => '[***]'`
	`  },`
	`  'StopAfterMatch' => '0'`
	`};`

6.5.22. PostMaster::PreFilterModule###3-NewTicketReject

Description:	Blocks all the incoming emails that do not have a valid ticket number in subject with From: @example.com address.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	`$Self->{'PostMaster::PreFilterModule'}->{'3-NewTicketReject'} =  {`
	`  'Match' => {`
	`    'From' => '@example.com'`
	`  },`
	`  'Module' => 'Kernel::System::PostMaster::Filter::NewTicketReject',`
	`  'Set' => {`
	`    'X-OTRS-Ignore' => 'yes'`
	`  },`
	`  'StopAfterMatch' => '0'`
	`};`

6.5.23. PostMaster::PreFilterModule::NewTicketReject::Sender

Description:	Defines the sender for rejected emails.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	`$Self->{'PostMaster::PreFilterModule::NewTicketReject::Sender'} =`
	`'noreply@example.com';`

6.5.24. PostMaster::PreFilterModule::NewTicketReject::Subject

Description:	Defines the subject for rejected emails.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1

Config-Setting:	`$Self->{'PostMaster::PreFilterModule::NewTicketReject::Subject'} =`
	`'Email Rejected';`

6.5.25. PostMaster::PreFilterModule::NewTicketReject::Body

Description:	Defines the body text for rejected emails.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1

Config-Setting:	`$Self->{'PostMaster::PreFilterModule::NewTicketReject::Body'} =  '`
	`Dear Customer,`
	
	`Unfortunately we could not detect a valid ticket number`
	`in your subject, so this email can\'t be processed.`
	
	`Please create a new ticket via the customer panel.`
	
	`Thanks for your help!`
	
	` Your Helpdesk Team`
	`';`

6.5.26. PostMaster::PreFilterModule###4-CMD

Description:	CMD example setup. Ignores emails where external CMD returns some output on STDOUT (email will be piped into STDIN of some.bin).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	`$Self->{'PostMaster::PreFilterModule'}->{'4-CMD'} =  {`
	`  'CMD' => '/usr/bin/some.bin',`

	'Module' => 'Kernel::System::PostMaster::Filter::CMD',
	'Set' => {
	'X-OTRS-Ignore' => 'yes'
	}
	};

6.5.27. PostMaster::PreFilterModule###5-SpamAssassin

Description:	Spam Assassin example setup. Ignores emails that are marked with SpamAssassin.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	$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'
	}
	};

6.5.28. PostMaster::PreFilterModule###6-SpamAssassin

Description:	Spam Assassin example setup. Moves marked mails to spam queue.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	$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'
	}
	};

6.5.29. PostMaster::PreFilterModule###000-MatchDBSource

Description:	Module to use database filter storage.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1

Config-Setting:	$Self->{'PostMaster::PreFilterModule'}->{'000-MatchDBSource'} = {
	'Module' => 'Kernel::System::PostMaster::Filter::MatchDBSource'
	};

6.5.30. PostMaster::PostFilterModule###000-FollowUpArticleTypeCheck

Description:	Module to check if arrived emails should be marked as email-internal (because of original forwarded internal email). ArticleType and SenderType define the values for the arrived email/article.

Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	0

Config-Setting:	$Self->{'PostMaster::PostFilterModule'}->{'000-
	FollowUpArticleTypeCheck'} = {
	  'ArticleType' => 'email-internal',
	  'Module' =>
	 'Kernel::System::PostMaster::Filter::FollowUpArticleTypeCheck',
	  'SenderType' => 'customer'
	};

6.5.31. PostMaster::PreFilterModule###000-ExternalTicketNumberRecognition1

Description:	Recognize if a ticket is a follow up to an existing ticket using an external ticket number.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	$Self->{'PostMaster::PreFilterModule'}->{'000-
	ExternalTicketNumberRecognition1'} = {
	  'ArticleType' => 'note-report',
	  'DynamicFieldName' => 'Name_X',
	  'FromAddressRegExp' => '\\s*@example.com',
	  'Module' =>
	 'Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition',
	  'Name' => 'Some Description',
	  'NumberRegExp' => '\\s*Incident-(\\d.*)\\s*',
	  'SearchInBody' => '1',
	  'SearchInSubject' => '1',
	  'SenderType' => 'system',
	  'TicketStateTypes' => 'new;open'
	};

6.5.32. PostMaster::PreFilterModule###000-ExternalTicketNumberRecognition2

Description:	Recognize if a ticket is a follow up to an existing ticket using an external ticket number.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	$Self->{'PostMaster::PreFilterModule'}->{'000-
	ExternalTicketNumberRecognition2'} = {
	  'ArticleType' => 'note-report',
	  'DynamicFieldName' => 'Name_X',
	  'FromAddressRegExp' => '\\s*@example.com',
	  'Module' =>
	 'Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition',
	  'Name' => 'Some Description',
	  'NumberRegExp' => '\\s*Incident-(\\d.*)\\s*',
	  'SearchInBody' => '1',
	  'SearchInSubject' => '1',
	  'SenderType' => 'system',

	'TicketStateTypes' => 'new;open' };

6.5.33. PostMaster::PreFilterModule###000-ExternalTicketNumberRecognition3

Description:	Recognize if a ticket is a follow up to an existing ticket using an external ticket number.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	$Self->{'PostMaster::PreFilterModule'}->{'000-
	ExternalTicketNumberRecognition3'} = {
	  'ArticleType' => 'note-report',
	  'DynamicFieldName' => 'Name_X',
	  'FromAddressRegExp' => '\\s*@example.com',
	  'Module' =>
	 'Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition',
	  'Name' => 'Some Description',
	  'NumberRegExp' => '\\s*Incident-(\\d.*)\\s*',
	  'SearchInBody' => '1',
	  'SearchInSubject' => '1',
	  'SenderType' => 'system',
	  'TicketStateTypes' => 'new;open'
	};

6.5.34. PostMaster::PreFilterModule###000-ExternalTicketNumberRecognition4

Description:	Recognize if a ticket is a follow up to an existing ticket using an external ticket number.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	$Self->{'PostMaster::PreFilterModule'}->{'000-
	ExternalTicketNumberRecognition4'} = {
	  'ArticleType' => 'note-report',
	  'DynamicFieldName' => 'Name_X',
	  'FromAddressRegExp' => '\\s*@example.com',
	  'Module' =>
	 'Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition',
	  'Name' => 'Some Description',
	  'NumberRegExp' => '\\s*Incident-(\\d.*)\\s*',
	  'SearchInBody' => '1',
	  'SearchInSubject' => '1',
	  'SenderType' => 'system',
	  'TicketStateTypes' => 'new;open'
	};

6.5.35. SendNoAutoResponseRegExp

Description:	If this regex matches, no message will be send by the autoresponder.
Group:	Ticket
SubGroup:	Core::PostMaster

<table>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>1</td></tr>
<tr><td>Config-Setting:</td><td>$Self->{'SendNoAutoResponseRegExp'} = '(MAILER-DAEMON|postmaster|abuse)@.+?\\..+?';</td></tr>
</table>

6.5.36. AutoResponseForWebTickets

Description:	If this option is set to 'Yes', tickets created via the web interface, via Customers or Agents, will receive an autoresponse if configured. If this option is set to 'No', no autoresponses will be sent.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	$Self->{'AutoResponseForWebTickets'} = '1';

6.6. Core::Stats

6.6.1. Stats::DynamicObjectRegistration###Ticket

Description:	Module to generate ticket statistics.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0

Config-Setting:	$Self->{'Stats::DynamicObjectRegistration'}->{'Ticket'} = {
	  'Module' => 'Kernel::System::Stats::Dynamic::Ticket'
	};

6.6.2. Stats::DynamicObjectRegistration###TicketList

Description:	Determines if the statistics module may generate ticket lists.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0

Config-Setting:	$Self->{'Stats::DynamicObjectRegistration'}->{'TicketList'} = {
	  'Module' => 'Kernel::System::Stats::Dynamic::TicketList'
	};

6.6.3. Stats::DynamicObjectRegistration###TicketAccountedTime

Description:	Module to generate accounted time ticket statistics.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1

Required:	0

Config-Setting:	`$Self->{'Stats::DynamicObjectRegistration'}->{'TicketAccountedTime'}`
	`=  {`
	`  'Module' => 'Kernel::System::Stats::Dynamic::TicketAccountedTime'`
	`};`

6.6.4. Stats::DynamicObjectRegistration###TicketSolutionResponseTime

Description:	Module to generate ticket solution and response time statistics.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Stats::DynamicObjectRegistration'}-`
	`>{'TicketSolutionResponseTime'} =  {`
	`  'Module' =>`
	`'Kernel::System::Stats::Dynamic::TicketSolutionResponseTime'`
	`};`

6.7. Core::Ticket

6.7.1. Ticket::Hook

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:	`$Self->{'Ticket::Hook'} =  'Ticket#';`

6.7.2. Ticket::HookDivider

Description:	The divider between TicketHook and ticket number. E.g ': '.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::HookDivider'} =  '';`

6.7.3. Ticket::SubjectSize

Description:	Max size of the subjects in an email reply.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::SubjectSize'} =  '100';`

6.7.4. Ticket::SubjectRe

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:	`$Self->{'Ticket::SubjectRe'} =  'Re';`

6.7.5. Ticket::SubjectFwd

Description:	The text at the beginning of the subject when an email is forwarded, e.g. FW, Fwd, or WG.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::SubjectFwd'} =  'Fwd';`

6.7.6. Ticket::SubjectFormat

Description:	The format of the subject. 'Left' means '[TicketHook#:12345] Some Subject', 'Right' means 'Some Subject [TicketHook#:12345]', 'None' means 'Some Subject' and no ticket number. In the last case you should enable PostmasterFollowupSearchInRaw or PostmasterFollowUpSearchInReferences to recognize followups based on email headers and/or body.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::SubjectFormat'} =  'Left';`

6.7.7. Ticket::MergeDynamicFields

Description:	A list of dynamic fields that are merged into the main ticket during a merge operation. Only dynamic fields that are empty in the main ticket will be set.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::MergeDynamicFields'} =  [];`

6.7.8. Ticket::CustomQueue

Description:	Name of custom queue. The custom queue is a queue selection of your preferred queues and can be selected in the preferences settings.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomQueue'} =  'My Queues';`

6.7.9. Ticket::CustomService

Description:	Name of custom service. The custom service is a service selection of your preferred services and can be selected in the preferences settings.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomService'} =  'My Services';`

6.7.10. Ticket::NewArticleIgnoreSystemSender

Description:	Ignore article with system sender type for new article feature (e. g. auto responses or email notifications).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::NewArticleIgnoreSystemSender'} =  '0';`

6.7.11. Ticket::ChangeOwnerToEveryone

Description:	Changes the owner of tickets to everyone (useful for ASP). Normally only agent with rw permissions in the queue of the ticket will be shown.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ChangeOwnerToEveryone'} =  '0';`

6.7.12. Ticket::Responsible

Description:	Enables ticket responsible feature, to keep track of a specific ticket.

Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Responsible'} =  '0';`

6.7.13. Ticket::ResponsibleAutoSet

Description:	Automatically sets the owner of a ticket as the responsible for it (if ticket responsible feature is enabled).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::ResponsibleAutoSet'} =  '1';`

6.7.14. Ticket::Type

Description:	Allows defining new types for ticket (if ticket type feature is enabled).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Type'} =  '0';`

6.7.15. Ticket::Service

Description:	Allows defining services and SLAs for tickets (e. g. email, desktop, network, ...), and escalation attributes for SLAs (if ticket service/SLA feature is enabled).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Service'} =  '0';`

6.7.16. Ticket::Service::Default::UnknownCustomer

Description:	Allows default services to be selected also for non existing customers.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Service::Default::UnknownCustomer'} =  '0';`

6.7.17. Ticket::ArchiveSystem

Description:	Activates the ticket archive system to have a faster system by moving some tickets out of the daily scope. To search for these tickets, the archive flag has to be enabled in the ticket search.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ArchiveSystem'} =  '0';`

6.7.18. Ticket::ArchiveSystem::RemoveSeenFlags

Description:	Controls if the ticket and article seen flags are removed when a ticket is archived.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ArchiveSystem::RemoveSeenFlags'} =  '1';`

6.7.19. Ticket::ArchiveSystem::RemoveTicketWatchers

Description:	Removes the ticket watcher information when a ticket is archived.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ArchiveSystem::RemoveTicketWatchers'} =  '1';`

6.7.20. Ticket::CustomerArchiveSystem

Description:	Activates the ticket archive system search in the customer interface.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomerArchiveSystem'} =  '0';`

6.7.21. Ticket::NumberGenerator

Description:	Selects the ticket number generator module. "AutoIncrement" increments the ticket number, the SystemID and the counter are used with SystemID.counter format (e.g. 1010138, 1010139).

	With "Date" the ticket numbers will be generated by the current date, the SystemID and the counter. The format looks like Year.Month.Day.SystemID.counter (e.g. 200206231010138, 200206231010139). With "DateChecksum" the counter will be appended as checksum to the string of date and SystemID. The checksum will be rotated on a daily basis. The format looks like Year.Month.Day.SystemID.Counter.CheckSum (e.g. 2002070110101520, 2002070110101535). "Random" generates randomized ticket numbers in the format "SystemID.Random" (e.g. 100057866352, 103745394596).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::NumberGenerator'} =`
	`'Kernel::System::Ticket::Number::DateChecksum';`

6.7.22. Ticket::NumberGenerator::CheckSystemID

Description:	Checks the SystemID in ticket number detection for follow-ups (use "No" if SystemID has been changed after using the system).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::NumberGenerator::CheckSystemID'} =  '1';`

6.7.23. Ticket::NumberGenerator::MinCounterSize

Description:	Sets the minimal ticket counter size (if "AutoIncrement" was selected as TicketNumberGenerator). Default is 5, this means the counter starts from 10000.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::NumberGenerator::MinCounterSize'} =  '5';`

6.7.24. Ticket::NumberGenerator::Date::UseFormattedCounter

Description:	Enables the minimal ticket counter size (if "Date" was selected as TicketNumberGenerator).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::NumberGenerator::Date::UseFormattedCounter'} =  '0';`

6.7.25. Ticket::CounterLog

Description:	Log file for the ticket counter.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CounterLog'} = '<OTRS_CONFIG_Home>/var/log/TicketCounter.log';`

6.7.26. Ticket::IndexModule

Description:	IndexAccelerator: to choose your backend TicketViewAccelerator module. "RuntimeDB" generates each queue view on the fly from ticket table (no performance problems up to approx. 60.000 tickets in total and 6.000 open tickets in the system). "StaticDB" is the most powerful module, it uses an extra ticket-index table that works like a view (recommended if more than 80.000 and 6.000 open tickets are stored in the system). Use the script "bin/otrs.RebuildTicketIndex.pl" for initial index update.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::IndexModule'} = 'Kernel::System::Ticket::IndexAccelerator::RuntimeDB';`

6.7.27. Ticket::StorageModule

Description:	Saves the attachments of articles. "DB" stores all data in the database (not recommended for storing big attachments). "FS" stores the data on the filesystem; this is faster but the webserver should run under the OTRS user. You can switch between the modules even on a system that is already in production without any loss of data.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::StorageModule'} = 'Kernel::System::Ticket::ArticleStorageDB';`

6.7.28. Ticket::StorageModule::CheckAllBackends

Description:	Specifies whether all storage backends should be checked when looking for attachements. This is only required for installations where some attachements are in the file system, and others in the database.

Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::StorageModule::CheckAllBackends'} =  '0';`

6.7.29. ArticleDir

Description:	Specifies the directory to store the data in, if "FS" was selected for TicketStorageModule.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'ArticleDir'} =  '<OTRS_CONFIG_Home>/var/article';`

6.7.30. Ticket::EventModulePost###100-ArchiveRestore

Description:	Restores a ticket from the archive (only if the event is a state change, from closed to any open available state).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'100-ArchiveRestore'} =  {`
	`  'Event' => 'TicketStateUpdate',`
	`  'Module' => 'Kernel::System::Ticket::Event::ArchiveRestore'`
	`};`

6.7.31. Ticket::EventModulePost###110-AcceleratorUpdate

Description:	Updates the ticket index accelerator.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'110-AcceleratorUpdate'} =  {`
	`  'Event' => 'TicketStateUpdate|TicketQueueUpdate|TicketLockUpdate',`
	`  'Module' => 'Kernel::System::Ticket::Event::TicketAcceleratorUpdate'`
	`};`

6.7.32. Ticket::EventModulePost###120-ForceOwnerResetOnMove

Description:	Resets and unlocks the owner of a ticket if it was moved to another queue.
Group:	Ticket
SubGroup:	Core::Ticket

Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'120-ForceOwnerResetOnMove'} =`
	`{`
	`  'Event' => 'TicketQueueUpdate',`
	`  'Module' => 'Kernel::System::Ticket::Event::ForceOwnerReset'`
	`};`

6.7.33. Ticket::EventModulePost###130-ForceStateChangeOnLock

Description:	Forces to choose a different ticket state (from current) after lock action. Define the current state as key, and the next state after lock action as content.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'130-ForceStateChangeOnLock'} =`
	`{`
	`  'Event' => 'TicketLockUpdate',`
	`  'Module' => 'Kernel::System::Ticket::Event::ForceState',`
	`  'new' => 'open'`
	`};`

6.7.34. Ticket::EventModulePost###140-ResponsibleAutoSet

Description:	Automatically sets the responsible of a ticket (if it is not set yet) after the first owner update.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'140-ResponsibleAutoSet'} =  {`
	`  'Event' => 'TicketOwnerUpdate',`
	`  'Module' => 'Kernel::System::Ticket::Event::ResponsibleAutoSet'`
	`};`

6.7.35. Ticket::EventModulePost###150-TicketPendingTimeReset

Description:	Sets the PendingTime of a ticket to 0 if the state is changed to a non-pending state.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'150-TicketPendingTimeReset'} =`
	`{`
	`  'Event' => 'TicketStateUpdate',`
	`  'Module' => 'Kernel::System::Ticket::Event::TicketPendingTimeReset'`
	`};`

6.7.36. Ticket::EventModulePost###500-NotificationEvent

Description:	Sends the notifications which are configured in the admin interface under "Notfication (Event)".
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'500-NotificationEvent'} =  {`
	`  'Event' => '',`
	`  'Module' => 'Kernel::System::Ticket::Event::NotificationEvent',`
	`  'Transaction' => '1'`
	`};`

6.7.37. Ticket::EventModulePost###900-EscalationIndex

Description:	Updates the ticket escalation index after a ticket attribute got updated.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'900-EscalationIndex'} =  {`
	`  'Event' => 'TicketSLAUpdate|TicketQueueUpdate|TicketStateUpdate|`
	`TicketCreate|ArticleCreate',`
	`  'Module' => 'Kernel::System::Ticket::Event::TicketEscalationIndex'`
	`};`

6.7.38. Ticket::EventModulePost###900-EscalationStopEvents

Description:	Ticket event module that triggers the escalation stop events.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'900-EscalationStopEvents'} =  {`
	`  'Event' => 'TicketSLAUpdate|TicketQueueUpdate|TicketStateUpdate|`
	`ArticleCreate',`
	`  'Module' =>`
	` 'Kernel::System::Ticket::Event::TriggerEscalationStopEvents'`
	`};`

6.7.39. Ticket::EventModulePost###910-ForceUnlockOnMove

Description:	Forces to unlock tickets after being moved to another queue.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'910-ForceUnlockOnMove'} =  {`
	`  'Event' => 'TicketQueueUpdate',`
	`  'Module' => 'Kernel::System::Ticket::Event::ForceUnlock'`
	`};`

6.7.40. Ticket::EventModulePost###920-TicketArticleNewMessageUpdate

Description:	Update Ticket "Seen" flag if every article got seen or a new Article got created.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'920-`
	`TicketArticleNewMessageUpdate'} =  {`
	`  'Event' => 'ArticleCreate|ArticleFlagSet',`
	`  'Module' => 'Kernel::System::Ticket::Event::TicketNewMessageUpdate'`
	`};`

6.7.41. DynamicFieldFromCustomerUser::Mapping

Description:	Define a mapping between variables of the customer user data (keys) and dynamic fields of a ticket (values). The purpose is to store customer user data in ticket dynamic fields. The dynamic fields must be present in the system and should be enabled for AgentTicketFreeText, so that they can be set/updated manually by the agent. They mustn't be enabled for AgentTicketPhone, AgentTicketEmail and AgentTicketCustomer. If they were, they would have precedence over the automatically set values. To use this mapping, you have to also activate the next setting below.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0

Config-Setting:	`$Self->{'DynamicFieldFromCustomerUser::Mapping'} =  {`
	`  'UserFirstname' => 'CustomerFirstname'`
	`};`

6.7.42. Ticket::EventModulePost###930-DynamicFieldFromCustomerUser

Description:	This event module stores attributes from CustomerUser as DynamicFields tickets. Please see the setting above for how to configure the mapping.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'930-`
	`DynamicFieldFromCustomerUser'} =  {`
	`  'Event' => '(TicketCreate|TicketCustomerUpdate)',`

	`  'Module' =>`
	` 'Kernel::System::Ticket::Event::DynamicFieldFromCustomerUser'`
	`};`

6.7.43. Ticket::CustomModule###001-CustomModule

Description:	Overloads (redefines) existing functions in Kernel::System::Ticket. Used to easily add customizations.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0

Config-Setting:	` $Self->{'Ticket::CustomModule'}->{'001-CustomModule'} =`
	` 'Kernel::System::Ticket::Custom';`

6.7.44. Ticket::ViewableSenderTypes

Description:	Defines the default viewable sender types of a ticket (default: customer).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Config-Setting:	` $Self->{'Ticket::ViewableSenderTypes'} =  [`
	`  '\'customer\''`
	`];`

6.7.45. Ticket::ViewableLocks

Description:	Defines the viewable locks of a ticket. Default: unlock, tmp_lock.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Config-Setting:	` $Self->{'Ticket::ViewableLocks'} =  [`
	`  '\'unlock\'',`
	`  '\'tmp_lock\''`
	`];`

6.7.46. Ticket::ViewableStateType

Description:	Defines the valid state types for a ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Config-Setting:	` $Self->{'Ticket::ViewableStateType'} =  [`
	`  'new',`

	'open', 'pending reminder', 'pending auto'];

6.7.47. Ticket::UnlockStateType

Description:	Defines the valid states for unlocked tickets. To unlock tickets the script "bin/otrs.UnlockTickets.pl" can be used.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::UnlockStateType'} = ['new', 'open'];

6.7.48. Ticket::PendingNotificationOnlyToOwner

Description:	Sends reminder notifications of unlocked ticket after reaching the reminder date (only sent to ticket owner).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::PendingNotificationOnlyToOwner'} = '0';

6.7.49. Ticket::PendingNotificationNotToResponsible

Description:	Disables sending reminder notifications to the responsible agent of a ticket (Ticket::Responsible needs to be activated).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::PendingNotificationNotToResponsible'} = '0';

6.7.50. Ticket::PendingReminderStateType

Description:	Defines the state type of the reminder for pending tickets.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::PendingReminderStateType'} = ['pending reminder'];

6.7.51. Ticket::PendingAutoStateType

Description:	Determines the possible states for pending tickets that changed state after reaching time limit.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Config-Setting:	` $Self->{'Ticket::PendingAutoStateType'} =  [`
	`   'pending auto'`
	`];`

6.7.52. Ticket::StateAfterPending

Description:	Defines which states should be set automatically (Content), after the pending time of state (Key) has been reached.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Config-Setting:	` $Self->{'Ticket::StateAfterPending'} =  {`
	`   'pending auto close+' => 'closed successful',`
	`   'pending auto close-' => 'closed unsuccessful'`
	`};`

6.7.53. System::Permission

Description:	Standard available permissions for agents within the application. If more permissions are needed, they can be entered here. Permissions must be defined to be effective. Some other good permissions have also been provided built-in: note, close, pending, customer, freetext, move, compose, responsible, forward, and bounce. Make sure that "rw" is always the last registered permission.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Config-Setting:	` $Self->{'System::Permission'} =  [`
	`   'ro',`
	`   'move_into',`
	`   'create',`
	`   'note',`
	`   'owner',`
	`   'priority',`
	`   'rw'`
	`];`

6.7.54. Ticket::Permission###1-OwnerCheck

Description:	Module to check the owner of a ticket.

Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Permission'}->{'1-OwnerCheck'} =  {`
	`  'Granted' => '1',`
	`  'Module' => 'Kernel::System::Ticket::Permission::OwnerCheck',`
	`  'Required' => '0'`
	`};`

6.7.55. Ticket::Permission###2-ResponsibleCheck

Description:	Module to check the agent responsible of a ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Permission'}->{'2-ResponsibleCheck'} =  {`
	`  'Granted' => '1',`
	`  'Module' => 'Kernel::System::Ticket::Permission::ResponsibleCheck',`
	`  'Required' => '0'`
	`};`

6.7.56. Ticket::Permission###3-GroupCheck

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:	`$Self->{'Ticket::Permission'}->{'3-GroupCheck'} =  {`
	`  'Granted' => '1',`
	`  'Module' => 'Kernel::System::Ticket::Permission::GroupCheck',`
	`  'Required' => '0'`
	`};`

6.7.57. Ticket::Permission###4-WatcherCheck

Description:	Module to check the watcher agents of a ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Permission'}->{'4-WatcherCheck'} =  {`
	`  'Granted' => '1',`
	`  'Module' => 'Kernel::System::Ticket::Permission::WatcherCheck',`
	`  'Required' => '0'`
	`};`

6.7.58. CustomerTicket::Permission###1-GroupCheck

Description:	Module to check the group permissions for the access to customer tickets.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	$Self->{'CustomerTicket::Permission'}->{'1-GroupCheck'} = {
	  'Granted' => '0',
	  'Module' =>
	 'Kernel::System::Ticket::CustomerPermission::GroupCheck',
	  'Required' => '1'
	};

6.7.59. CustomerTicket::Permission###2-CustomerUserIDCheck

Description:	Grants access, if the customer ID of the ticket matches the customer user's ID and the customer user has group permissions on the queue the ticket is in.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	$Self->{'CustomerTicket::Permission'}->{'2-CustomerUserIDCheck'} = {
	  'Granted' => '1',
	  'Module' =>
	 'Kernel::System::Ticket::CustomerPermission::CustomerUserIDCheck',
	  'Required' => '0'
	};

6.7.60. CustomerTicket::Permission###3-CustomerIDCheck

Description:	Module to check customer permissions.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	$Self->{'CustomerTicket::Permission'}->{'3-CustomerIDCheck'} = {
	  'Granted' => '1',
	  'Module' =>
	 'Kernel::System::Ticket::CustomerPermission::CustomerIDCheck',
	  'Required' => '0'
	};

6.7.61. Ticket::DefineEmailFrom

Description:	Defines how the From field from the emails (sent from answers and email tickets) should look like.
Group:	Ticket
SubGroup:	Core::Ticket

Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::DefineEmailFrom'} =  'SystemAddressName';`

6.7.62. Ticket::DefineEmailFromSeparator

Description:	Defines the separator between the agents real name and the given queue email address.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::DefineEmailFromSeparator'} =  'via';`

6.7.63. CustomerNotifyJustToRealCustomer

Description:	Sends customer notifications just to the mapped customer. Normally, if no customer is mapped, the latest customer sender gets the notification.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerNotifyJustToRealCustomer'} =  '0';`

6.7.64. AgentSelfNotifyOnAction

Description:	Specifies if an agent should receive email notification of his own actions.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'AgentSelfNotifyOnAction'} =  '0';`

6.7.65. Ticket::EventModulePost###900-GenericAgent

Description:	Event module registration. For more performance you can define a trigger event (e. g. Event => TicketCreate).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'900-GenericAgent'} =  {`
	`  'Event' => '',`
	`  'Module' => 'Kernel::System::Ticket::Event::GenericAgent',`

	'Transaction' => '1' };

6.7.66. Ticket::UnlockOnAway

Description:	Unlock tickets whenever a note is added and the owner is out of office.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::UnlockOnAway'} = '0';

6.7.67. StandardTemplate::Types

Description:	Defines the list of types for templates.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	$Self->{'StandardTemplate::Types'} = {
	  'Answer' => 'Answer',
	  'Create' => 'Create',
	  'Forward' => 'Forward',
	  'Note' => 'Note',
	  'PhoneCall' => 'Phone call'
	};

6.8. Core::TicketACL

6.8.1. Ticket::Acl::Module###1-Ticket::Acl::Module

Description:	ACL module that allows closing parent tickets only if all its children are already closed ("State" shows which states are not available for the parent ticket until all child tickets are closed).
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	0
Required:	0

Config-Setting:	$Self->{'Ticket::Acl::Module'}->{'1-Ticket::Acl::Module'} = {
	  'Module' =>
	 'Kernel::System::Ticket::Acl::CloseParentAfterClosedChilds',
	  'State' => [
	    'closed successful',
	    'closed unsuccessful'
	  ]
	};

6.8.2. TicketACL::Default::Action

Description:	Default ACL values for ticket actions.
Group:	Ticket

SubGroup:	Core::TicketACL
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TicketACL::Default::Action'} =  {};`

6.8.3. ACLKeysLevel1Match

Description:	Defines which items are available in first level of the ACL structure.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	0

Config-Setting:	`$Self->{'ACLKeysLevel1Match'} =  {`
	`  'Properties' => 'Properties',`
	`  'PropertiesDatabase' => 'PropertiesDatabase'`
	`};`

6.8.4. ACLKeysLevel1Change

Description:	Defines which items are available in first level of the ACL structure.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	0

Config-Setting:	`$Self->{'ACLKeysLevel1Change'} =  {`
	`  'Possible' => 'Possible',`
	`  'PossibleAdd' => 'PossibleAdd',`
	`  'PossibleNot' => 'PossibleNot'`
	`};`

6.8.5. ACLKeysLevel2::Possible

Description:	Defines which items are available in second level of the ACL structure.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	0

Config-Setting:	`$Self->{'ACLKeysLevel2::Possible'} =  {`
	`  'Action' => 'Action',`
	`  'ActivityDialog' => 'ActivityDialog',`
	`  'Process' => 'Process',`
	`  'Ticket' => 'Ticket'`
	`};`

6.8.6. ACLKeysLevel2::PossibleAdd

Description:	Defines which items are available in second level of the ACL structure.

Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	0

Config-Setting:	`$Self->{'ACLKeysLevel2::PossibleAdd'} =  {`
	`  'Action' => 'Action',`
	`  'Ticket' => 'Ticket'`
	`};`

6.8.7. ACLKeysLevel2::PossibleNot

Description:	Defines which items are available in second level of the ACL structure.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	0

Config-Setting:	`$Self->{'ACLKeysLevel2::PossibleNot'} =  {`
	`  'Action' => 'Action',`
	`  'ActivityDialog' => 'ActivityDialog',`
	`  'Process' => 'Process',`
	`  'Ticket' => 'Ticket'`
	`};`

6.8.8. ACLKeysLevel2::Properties

Description:	Defines which items are available in second level of the ACL structure.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	0

Config-Setting:	`$Self->{'ACLKeysLevel2::Properties'} =  {`
	`  'CustomerUser' => 'CustomerUser',`
	`  'DynamicField' => 'DynamicField',`
	`  'Frontend' => 'Frontend',`
	`  'Owner' => 'Owner',`
	`  'Priority' => 'Priority',`
	`  'Process' => 'Process',`
	`  'Queue' => 'Queue',`
	`  'Responsible' => 'Responsible',`
	`  'SLA' => 'SLA',`
	`  'Service' => 'Service',`
	`  'State' => 'State',`
	`  'Ticket' => 'Ticket',`
	`  'Type' => 'Type',`
	`  'User' => 'User'`
	`};`

6.8.9. ACLKeysLevel2::PropertiesDatabase

Description:	Defines which items are available in second level of the ACL structure.

Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	0

Config-Setting:	`$Self->{'ACLKeysLevel2::PropertiesDatabase'} =  {`
	`  'CustomerUser' => 'CustomerUser',`
	`  'DynamicField' => 'DynamicField',`
	`  'Owner' => 'Owner',`
	`  'Priority' => 'Priority',`
	`  'Process' => 'Process',`
	`  'Queue' => 'Queue',`
	`  'Responsible' => 'Responsible',`
	`  'SLA' => 'SLA',`
	`  'Service' => 'Service',`
	`  'State' => 'State',`
	`  'Ticket' => 'Ticket',`
	`  'Type' => 'Type',`
	`  'User' => 'User'`
	`};`

6.8.10. ACLKeysLevel3::Actions###100-Default

Description:	Defines which items are available for 'Action' in third level of the ACL structure.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	0

Config-Setting:	`$Self->{'ACLKeysLevel3::Actions'}->{'100-Default'} =  [`
	`  'AgentTicketBounce',`
	`  'AgentTicketClose',`
	`  'AgentTicketCompose',`
	`  'AgentTicketCustomer',`
	`  'AgentTicketForward',`
	`  'AgentTicketFreeText',`
	`  'AgentTicketHistory',`
	`  'AgentTicketLink',`
	`  'AgentTicketLock',`
	`  'AgentTicketMerge',`
	`  'AgentTicketMove',`
	`  'AgentTicketNote',`
	`  'AgentTicketOwner',`
	`  'AgentTicketPending',`
	`  'AgentTicketPhone',`
	`  'AgentTicketPhoneInbound',`
	`  'AgentTicketPhoneOutbound',`
	`  'AgentTicketPrint',`
	`  'AgentTicketPriority',`
	`  'AgentTicketResponsible',`
	`  'AgentTicketWatcher',`
	`  'AgentTicketZoom',`
	`  'AgentLinkObject'`
	`];`

6.8.11. ACL::CacheTTL

Description:	Cache time in seconds for the DB ACL backend.

Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	1
Config-Setting:	`$Self->{'ACL::CacheTTL'} =  '3600';`

6.8.12. TicketACL::Debug::Enabled

Description:	If enabled debugging information for ACLs is logged.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TicketACL::Debug::Enabled'} =  '0';`

6.8.13. TicketACL::Debug::LogPriority

Description:	Defines the priority in which the information is logged and presented.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	0
Required:	0
Config-Setting:	`$Self->{'TicketACL::Debug::LogPriority'} =  'debug';`

6.8.14. TicketACL::Debug::Filter###00-Default

Description:	Filter for debugging ACLs. Note: More ticket attributes can be added in the format <OTRS_TICKET_Attribute> e.g. <OTRS_TICKET_Priority>.
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	0
Required:	0

Config-Setting:	`$Self->{'TicketACL::Debug::Filter'}->{'00-Default'} =  {`
	`  '<OTRS_TICKET_TicketNumber>' => '',`
	`  'ACLName' => ''`
	`};`

6.9. Core::TicketBulkAction

6.9.1. Ticket::Frontend::BulkFeature

Description:	Enables ticket bulk action feature for the agent frontend to work on more than one ticket at a time.
Group:	Ticket
SubGroup:	Core::TicketBulkAction

Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::BulkFeature'} =  '1';`

6.9.2. Ticket::Frontend::BulkFeatureGroup

Description:	Enables ticket bulk action feature only for the listed groups.
Group:	Ticket
SubGroup:	Core::TicketBulkAction
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::BulkFeatureGroup'} =  [`
	`  'admin',`
	`  'users'`
	`];`

6.10. Core::TicketDynamicFieldDefault

6.10.1. Ticket::EventModulePost###TicketDynamicFieldDefault

Description:	Event module registration. For more performance you can define a trigger event (e. g. Event => TicketCreate). This is only possible if all Ticket dynamic fields need the same event.
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'TicketDynamicFieldDefault'} =`
	`{`
	`  'Module' =>`
	`'Kernel::System::Ticket::Event::TicketDynamicFieldDefault',`
	`  'Transaction' => '1'`
	`};`

6.10.2. Ticket::TicketDynamicFieldDefault###Element1

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element1'} =  {`
	`  'Event' => 'TicketCreate',`
	`  'Name' => 'Field1',`
	`  'Value' => 'Default'`

	`};`

6.10.3. Ticket::TicketDynamicFieldDefault###Element2

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element2'} =  {`
	`  'Event' => '',`
	`  'Name' => '',`
	`  'Value' => ''`
	`};`

6.10.4. Ticket::TicketDynamicFieldDefault###Element3

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element3'} =  {`
	`  'Event' => '',`
	`  'Name' => '',`
	`  'Value' => ''`
	`};`

6.10.5. Ticket::TicketDynamicFieldDefault###Element4

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element4'} =  {`
	`  'Event' => '',`
	`  'Name' => '',`
	`  'Value' => ''`

	};

6.10.6. Ticket::TicketDynamicFieldDefault###Element5

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element5'} = {
	  'Event' => '',
	  'Name' => '',
	  'Value' => ''
	};

6.10.7. Ticket::TicketDynamicFieldDefault###Element6

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element6'} = {
	  'Event' => '',
	  'Name' => '',
	  'Value' => ''
	};

6.10.8. Ticket::TicketDynamicFieldDefault###Element7

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element7'} = {
	  'Event' => '',
	  'Name' => '',
	  'Value' => ''

	};

6.10.9. Ticket::TicketDynamicFieldDefault###Element8

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element8'} = {
	  'Event' => '',
	  'Name' => '',
	  'Value' => ''
	};

6.10.10. Ticket::TicketDynamicFieldDefault###Element9

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element9'} = {
	  'Event' => '',
	  'Name' => '',
	  'Value' => ''
	};

6.10.11. Ticket::TicketDynamicFieldDefault###Element10

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element10'} = {
	  'Event' => '',
	  'Name' => '',
	  'Value' => ''

	};

6.10.12. Ticket::TicketDynamicFieldDefault###Element11

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element11'} = {
	'Event' => '',
	'Name' => '',
	'Value' => ''
	};

6.10.13. Ticket::TicketDynamicFieldDefault###Element12

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element12'} = {
	'Event' => '',
	'Name' => '',
	'Value' => ''
	};

6.10.14. Ticket::TicketDynamicFieldDefault###Element13

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element13'} = { 'Event' => '', 'Name' => '', 'Value' => ''

	`};`

6.10.15. Ticket::TicketDynamicFieldDefault###Element14

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element14'} =  {`
	`  'Event' => '',`
	`  'Name' => '',`
	`  'Value' => ''`
	`};`

6.10.16. Ticket::TicketDynamicFieldDefault###Element15

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element15'} =  {`
	`  'Event' => '',`
	`  'Name' => '',`
	`  'Value' => ''`
	`};`

6.10.17. Ticket::TicketDynamicFieldDefault###Element16

Description:	Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element16'} =  {`
	`  'Event' => '',`
	`  'Name' => '',`
	`  'Value' => ''`

	};

6.11. Core::TicketWatcher

6.11.1. Ticket::Watcher

Description:	Enables or disables the ticket watcher feature, to keep track of tickets without being the owner nor the responsible.
Group:	Ticket
SubGroup:	Core::TicketWatcher
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Watcher'} =  '0';`

6.11.2. Ticket::WatcherGroup

Description:	Enables ticket watcher feature only for the listed groups.
Group:	Ticket
SubGroup:	Core::TicketWatcher
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::WatcherGroup'} =  [`
	`  'admin',`
	`  'users'`
	`];`

6.12. Frontend::Admin

6.12.1. Events###Ticket

Description:	List of all ticket events to be displayed in the GUI.
Group:	Ticket
SubGroup:	Frontend::Admin
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Events'}->{'Ticket'} =  [`
	`  'TicketCreate',`
	`  'TicketDelete',`
	`  'TicketTitleUpdate',`
	`  'TicketUnlockTimeoutUpdate',`
	`  'TicketQueueUpdate',`
	`  'TicketTypeUpdate',`
	`  'TicketServiceUpdate',`
	`  'TicketSLAUpdate',`
	`  'TicketCustomerUpdate',`
	`  'TicketPendingTimeUpdate',`
	`  'TicketLockUpdate',`
	`  'TicketArchiveFlagUpdate',`
	`  'TicketStateUpdate',`
	`  'TicketOwnerUpdate',`
	`  'TicketResponsibleUpdate',`
	`  'TicketPriorityUpdate',`
	`  'HistoryAdd',`
	`  'HistoryDelete',`

	`'TicketAccountTime',`
	`'TicketMerge',`
	`'TicketSubscribe',`
	`'TicketUnsubscribe',`
	`'TicketFlagSet',`
	`'TicketFlagDelete',`
	`'TicketSlaveLinkAdd',`
	`'TicketSlaveLinkDelete',`
	`'TicketMasterLinkDelete',`
	`'EscalationResponseTimeNotifyBefore',`
	`'EscalationUpdateTimeNotifyBefore',`
	`'EscalationSolutionTimeNotifyBefore',`
	`'EscalationResponseTimeStart',`
	`'EscalationUpdateTimeStart',`
	`'EscalationSolutionTimeStart',`
	`'EscalationResponseTimeStop',`
	`'EscalationUpdateTimeStop',`
	`'EscalationSolutionTimeStop'`
	`];`

6.12.2. Events###Article

Description:	List of all article events to be displayed in the GUI.
Group:	Ticket
SubGroup:	Frontend::Admin
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Events'}->{'Article'} =  [`
	`'ArticleCreate',`
	`'ArticleUpdate',`
	`'ArticleSend',`
	`'ArticleBounce',`
	`'ArticleAgentNotification',`
	`'ArticleCustomerNotification',`
	`'ArticleAutoResponse',`
	`'ArticleFlagSet',`
	`'ArticleFlagDelete',`
	`'ArticleAgentNotification',`
	`'ArticleCustomerNotification'`
	`];`

6.13. Frontend::Admin::AdminNotificationEvent

6.13.1. Frontend::Admin::AdminNotificationEvent###RichText

Description:	Uses richtext for viewing and editing notification events.
Group:	Ticket
SubGroup:	Frontend::Admin::AdminNotificationEvent
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Admin::AdminNotificationEvent'}->{'RichText'} =`
	`'1';`

6.13.2. Frontend::Admin::AdminNotificationEvent###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Group:	Ticket
SubGroup:	Frontend::Admin::AdminNotificationEvent
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Admin::AdminNotificationEvent'}->{'RichTextWidth'} =  '620';`

6.13.3. Frontend::Admin::AdminNotificationEvent###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Admin::AdminNotificationEvent
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Admin::AdminNotificationEvent'}->{'RichTextHeight'} =  '320';`

6.14. Frontend::Admin::ModuleRegistration

6.14.1. Frontend::Module###AdminACL

<table>
<tr><td>Description:</td><td>Frontend module registration for the agent interface.</td></tr>
<tr><td>Group:</td><td>Ticket</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Admin::ModuleRegistration</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td><pre>
$Self->{'Frontend::Module'}->{'AdminACL'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ACL.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ACL.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Configure and manage ACLs.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Access Control Lists (ACL)',
    'Prio' => '750'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Access Control Lists (ACL)'
};
</pre></td></tr>
</table>

6.14.2. Frontend::Module###AdminQueue

Description:	Frontend module registration for the agent interface.

Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AdminQueue'} = {
	  'Description' => 'Admin',
	  'Group' => [
	    'admin'
	  ],
	  'NavBarModule' => {
	    'Block' => 'Queue',
	    'Description' => 'Create and manage queues.',
	    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	    'Name' => 'Queues',
	    'Prio' => '100'
	  },
	  'NavBarName' => 'Admin',
	  'Title' => 'Queues'
	};

6.14.3. Frontend::Module###AdminTemplate

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AdminTemplate'} = {
	  'Description' => 'Admin',
	  'Group' => [
	    'admin'
	  ],
	  'NavBarModule' => {
	    'Block' => 'Queue',
	    'Description' => 'Create and manage templates.',
	    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	    'Name' => 'Templates',
	    'Prio' => '200'
	  },
	  'NavBarName' => 'Admin',
	  'Title' => 'Templates'
	};

6.14.4. Frontend::Module###AdminQueueTemplates

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AdminQueueTemplates'} = {
	  'Description' => 'Admin',
	  'Group' => [
	    'admin'
	  ],

```
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Link templates to queues.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Templates <-> Queues',
    'Prio' => '300'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Templates <-> Queues'
};
```

6.14.5. Frontend::Module###AdminAutoResponse

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminAutoResponse'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Create and manage responses that are
 automatically sent.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Auto Responses',
    'Prio' => '400'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Auto Responses'
};
```

6.14.6. Frontend::Module###AdminQueueAutoResponse

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminQueueAutoResponse'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Link queues to auto responses.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Auto Responses <-> Queues',
    'Prio' => '500'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Auto Responses <-> Queues'
};
```

6.14.7. Frontend::Module###AdminAttachment

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminAttachment'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Create and manage attachments.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Attachments',
    'Prio' => '600'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Attachments'
};
```

6.14.8. Frontend::Module###AdminTemplateAttachment

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminTemplateAttachment'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Link attachments to templates.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Attachments <-> Templates',
    'Prio' => '700'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Attachments <-> Templates'
};
```

6.14.9. Frontend::Module###AdminSalutation

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminSalutation'} =  {
```

	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'NavBarModule' => {`
	`    'Block' => 'Queue',`
	`    'Description' => 'Create and manage salutations.',`
	`    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`    'Name' => 'Salutations',`
	`    'Prio' => '800'`
	`  },`
	`  'NavBarName' => 'Admin',`
	`  'Title' => 'Salutations'`
	`};`

6.14.10. Frontend::Module###AdminSignature

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	` $Self->{'Frontend::Module'}->{'AdminSignature'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'NavBarModule' => {`
	`    'Block' => 'Queue',`
	`    'Description' => 'Create and manage signatures.',`
	`    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`    'Name' => 'Signatures',`
	`    'Prio' => '900'`
	`  },`
	`  'NavBarName' => 'Admin',`
	`  'Title' => 'Signatures'`
	`};`

6.14.11. Frontend::Module###AdminSystemAddress

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	` $Self->{'Frontend::Module'}->{'AdminSystemAddress'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'NavBarModule' => {`
	`    'Block' => 'Email',`
	`    'Description' => 'Set sender email addresses for this system.',`
	`    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`    'Name' => 'Email Addresses',`
	`    'Prio' => '300'`
	`  },`
	`  'NavBarName' => 'Admin',`

```
  'Title' => 'Email Addresses'
};
```

6.14.12. Frontend::Module###AdminNotification

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminNotification'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Manage notifications that are sent to agents.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Agent Notifications',
    'Prio' => '400'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Agent Notifications'
};
```

6.14.13. Frontend::Module###AdminNotificationEvent

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AdminNotificationEvent'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Create and manage event based notifications.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Notifications (Event)',
    'Prio' => '400'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Notifications (Event)'
};
```

6.14.14. Frontend::Module###AdminService

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1

Required:	0

Config-Setting:

```
$Self->{'Frontend::Module'}->{'AdminService'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Create and manage services.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Services',
    'Prio' => '900'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Services'
};
```

6.14.15. Frontend::Module###AdminSLA

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
$Self->{'Frontend::Module'}->{'AdminSLA'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Create and manage Service Level Agreements
(SLAs).',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Service Level Agreements',
    'Prio' => '1000'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Service Level Agreements'
};
```

6.14.16. Frontend::Module###AdminType

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
$Self->{'Frontend::Module'}->{'AdminType'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Create and manage ticket types.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
```

	`    'Name' => 'Types',`
	`    'Prio' => '700'`
	`  },`
	`  'NavBarName' => 'Admin',`
	`  'Title' => 'Types'`
	`};`

6.14.17. Frontend::Module###AdminState

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminState'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'NavBarModule' => {`
	`    'Block' => 'Ticket',`
	`    'Description' => 'Create and manage ticket states.',`
	`    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`    'Name' => 'States',`
	`    'Prio' => '800'`
	`  },`
	`  'NavBarName' => 'Admin',`
	`  'Title' => 'States'`
	`};`

6.14.18. Frontend::Module###AdminPriority

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminPriority'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'NavBarModule' => {`
	`    'Block' => 'Ticket',`
	`    'Description' => 'Create and manage ticket priorities.',`
	`    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`    'Name' => 'Priorities',`
	`    'Prio' => '850'`
	`  },`
	`  'NavBarName' => 'Admin',`
	`  'Title' => 'Priorities'`
	`};`

6.14.19. Frontend::Module###AdminGenericAgent

Description:	Frontend module registration for the agent interface.

<table>
<tr><td>Group:</td><td>Ticket</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Admin::ModuleRegistration</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td><pre>
$Self->{'Frontend::Module'}->{'AdminGenericAgent'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.GenericAgent.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Manage tasks triggered by event or time based
execution.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'GenericAgent',
    'Prio' => '300'
  },
  'NavBarName' => 'Admin',
  'Title' => 'GenericAgent'
};
</pre></td></tr>
</table>

6.15. Frontend::Agent

6.15.1. Ticket::Frontend::PendingDiffTime

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:	`$Self->{'Ticket::Frontend::PendingDiffTime'} =  '86400';`

6.15.2. Ticket::Frontend::MaxQueueLevel

Description:	Define the max depth of queues.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::MaxQueueLevel'} =  '5';`

6.15.3. Ticket::Frontend::ListType

Description:	Shows existing parent/child queue lists in the system in the form of a tree or a list.
Group:	Ticket

SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ListType'} =  'tree';`

6.15.4. Ticket::Frontend::TextAreaEmail

Description:	Permitted width for compose email windows.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::TextAreaEmail'} =  '82';`

6.15.5. Ticket::Frontend::TextAreaNote

Description:	Permitted width for compose note windows.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::TextAreaNote'} =  '78';`

6.15.6. Ticket::Frontend::InformAgentMaxSize

Description:	Max size (in rows) of the informed agents box in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::InformAgentMaxSize'} =  '3';`

6.15.7. Ticket::Frontend::InvolvedAgentMaxSize

Description:	Max size (in rows) of the involved agents box in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::InvolvedAgentMaxSize'} =  '3';`

6.15.8. Ticket::Frontend::CustomerInfoCompose

Description:	Shows the customer user information (phone and email) in the compose screen.

Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerInfoCompose'} =  '1';`

6.15.9. Ticket::Frontend::CustomerInfoComposeMaxSize

Description:	Max size (in characters) of the customer information table (phone and email) in the compose screen.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerInfoComposeMaxSize'} =  '22';`

6.15.10. Ticket::Frontend::CustomerInfoZoom

Description:	Shows the customer user's info in the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerInfoZoom'} =  '1';`

6.15.11. Ticket::Frontend::CustomerInfoZoomMaxSize

Description:	Maximum size (in characters) of the customer information table in the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerInfoZoomMaxSize'} =  '22';`

6.15.12. Ticket::Frontend::DynamicFieldsZoomMaxSizeSidebar

Description:	Maximum length (in characters) of the dynamic field in the sidebar of the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::DynamicFieldsZoomMaxSizeSidebar'} =  '18';`

6.15.13. Ticket::Frontend::DynamicFieldsZoomMaxSizeArticle

Description:	Maximum length (in characters) of the dynamic field in the article of the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::DynamicFieldsZoomMaxSizeArticle'} = '160';`

6.15.14. Ticket::Frontend::AccountTime

Description:	Activates time accounting.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AccountTime'} =  '1';`

6.15.15. Ticket::Frontend::TimeUnits

Description:	Sets the prefered time units (e.g. work units, hours, minutes).
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::TimeUnits'} =  ' (work units)';`

6.15.16. Ticket::Frontend::NeedAccountedTime

Description:	Defines if time accounting is mandatory in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NeedAccountedTime'} =  '0';`

6.15.17. Ticket::Frontend::BulkAccountedTime

Description:	Defines if time accounting must be set to all tickets in bulk action.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::BulkAccountedTime'} =  '1';`

6.15.18. Ticket::Frontend::NeedSpellCheck

Description:	Defines if composed messages have to be spell checked in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NeedSpellCheck'} =  '0';`

6.15.19. Ticket::Frontend::NewOwnerSelection

Description:	Shows an owner selection in phone and email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NewOwnerSelection'} =  '1';`

6.15.20. Ticket::Frontend::NewResponsibleSelection

Description:	Show a responsible selection in phone and email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NewResponsibleSelection'} =  '1';`

6.15.21. Ticket::Frontend::NewQueueSelectionType

Description:	Defines the receipent target of the phone ticket and the sender of the email ticket ("Queue" shows all queues, "SystemAddress" displays all system addresses) in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NewQueueSelectionType'} =  'Queue';`

6.15.22. Ticket::Frontend::NewQueueSelectionString

Description:	Determines the strings that will be shown as receipent (To:) of the phone ticket and as sender (From:) of the email ticket in the agent

	interface. For Queue as NewQueueSelectionType "<Queue>" shows the names of the queues and for SystemAddress "<Realname> <<Email>>" shows the name and email of the receipent.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NewQueueSelectionString'} =  '<Queue>';`

6.15.23. Ticket::Frontend::NewQueueOwnSelection

Description:	Determines which options will be valid of the recepient (phone ticket) and the sender (email ticket) in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::NewQueueOwnSelection'} =  {`
	`  '1' => 'First Queue!',`
	`  '2' => 'Second Queue!'`
	`};`

6.15.24. Ticket::Frontend::ShowCustomerTickets

Description:	Shows customer history tickets in AgentTicketPhone, AgentTicketEmail and AgentTicketCustomer.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ShowCustomerTickets'} =  '1';`

6.15.25. NewTicketInNewWindow::Enabled

Description:	If enabled, TicketPhone and TicketEmail will be open in new windows.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'NewTicketInNewWindow::Enabled'} =  '0';`

6.15.26. CustomerDBLink

Description:	Defines an external link to the database of the customer (e.g. 'http://yourhost/customer.php?CID=[% Data.CustomerID %]' or '').
Group:	Ticket
SubGroup:	Frontend::Agent

Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerDBLink'} =  '[% Env("CGIHandle") %]?Action=AgentCustomerInformationCenter;CustomerID=[% Data.CustomerID | uri %]';`

6.15.27. CustomerDBLinkTarget

Description:	Defines the target attribute in the link to external customer database. E.g. 'target="cdb"'.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerDBLinkTarget'} =  '';`

6.15.28. CustomerDBLinkClass

Description:	Defines the target attribute in the link to external customer database. E.g. 'AsPopup PopupType_TicketAction'.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerDBLinkClass'} =  '';`

6.15.29. Frontend::CommonObject###QueueObject

Description:	Path of the file that stores all the settings for the QueueObject object for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::CommonObject'}->{'QueueObject'} = 'Kernel::System::Queue';`

6.15.30. Frontend::CommonObject###TicketObject

Description:	Path of the file that stores all the settings for the TicketObject for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::CommonObject'}->{'TicketObject'} = 'Kernel::System::Ticket';`

6.15.31. Frontend::CommonParam###Action

Description:	Defines the default used Frontend-Module if no Action parameter given in the url on the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::CommonParam'}->{'Action'} =  'AgentDashboard';`

6.15.32. Frontend::CommonParam###QueueID

Description:	Default queue ID used by the system in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::CommonParam'}->{'QueueID'} =  '0';`

6.15.33. Frontend::CommonParam###TicketID

Description:	Default ticket ID used by the system in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::CommonParam'}->{'TicketID'} =  '';`

6.15.34. DefaultOverviewColumns

Description:	General ticket data shown in the ticket overviews (fall-back). Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note that TicketNumber can not be disabled, because it is necessary.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1

Config-Setting:	`$Self->{'DefaultOverviewColumns'} =  {`
	`'Age' => '2',`
	`'Changed' => '1',`
	`'CustomerID' => '2',`
	`'CustomerName' => '1',`
	`'CustomerUserID' => '1',`
	`'EscalationResponseTime' => '1',`
	`'EscalationSolutionTime' => '1',`
	`'EscalationTime' => '1',`
	`'EscalationUpdateTime' => '1',`
	`'Lock' => '2',`

```
  'Owner' => '2',
  'PendingTime' => '1',
  'Priority' => '1',
  'Queue' => '2',
  'Responsible' => '1',
  'SLA' => '1',
  'Service' => '1',
  'State' => '2',
  'TicketNumber' => '2',
  'Title' => '2',
  'Type' => '1'
};
```

6.16. Frontend::Agent::Dashboard

6.16.1. DashboardBackend###0100-TicketPendingReminder

Description:	Parameters for the dashboard backend of the ticket pending reminder overview of the agent interface . "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'DashboardBackend'}->{'0100-TicketPendingReminder'} =  {
  'Attributes' => 'TicketPendingTimeOlderMinutes=1;StateType=pending
reminder;SortBy=PendingTime;OrderBy=Down;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All tickets with a reminder set where the reminder
date has been reached',
  'Filter' => 'Locked',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
  'Permission' => 'rw',
  'Time' => 'UntilTime',
  'Title' => 'Reminder Tickets'
};
```

6.16.2. DashboardBackend###0110-TicketEscalation

Description:	Parameters for the dashboard backend of the ticket escalation overview of the agent interface . "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'DashboardBackend'}->{'0110-TicketEscalation'} =  {
  'Attributes' =>
'TicketEscalationTimeOlderMinutes=1;SortBy=EscalationTime;OrderBy=Down;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All escalated tickets',
  'Filter' => 'All',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
  'Permission' => 'rw',
  'Time' => 'EscalationTime',
  'Title' => 'Escalated Tickets'
```

<table>
<tr><td></td><td><pre>};</pre></td></tr>
</table>

6.16.3. DashboardBackend###0120-TicketNew

<table>
<tr><td>Description:</td><td>Parameters for the dashboard backend of the new tickets overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.</td></tr>
<tr><td>Group:</td><td>Ticket</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Agent::Dashboard</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'DashboardBackend'}->{'0120-TicketNew'} = {
  'Attributes' => 'StateType=new;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All new tickets, these tickets have not been
 worked on yet',
  'Filter' => 'All',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
  'Permission' => 'rw',
  'Time' => 'Age',
  'Title' => 'New Tickets'
};
</pre></td></tr>
</table>

6.16.4. DashboardBackend###0130-TicketOpen

<table>
<tr><td>Description:</td><td>Parameters for the dashboard backend of the ticket pending reminder overview of the agent interface. "Limit" is the number</td></tr>
</table>

	of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:

```
 $Self->{'DashboardBackend'}->{'0130-TicketOpen'} =  {
  'Attributes' => 'StateType=open;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All open tickets, these tickets have already been
 worked on, but need a response',
  'Filter' => 'All',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
  'Permission' => 'rw',
  'Time' => 'Age',
  'Title' => 'Open Tickets / Need to be answered'
};
```

6.16.5. DashboardBackend###0250-TicketStats

Description:	Parameters for the dashboard backend of the ticket stats of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket

SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DashboardBackend'}->{'0250-TicketStats'} =  {`
	`  'Block' => 'ContentSmall',`
	`  'CacheTTLLocal' => '30',`
	`  'Changed' => '1',`
	`  'Closed' => '1',`
	`  'Default' => '1',`
	`  'Group' => '',`
	`  'Module' => 'Kernel::Output::HTML::DashboardTicketStatsGeneric',`
	`  'Permission' => 'rw',`
	`  'Title' => '7 Day Stats'`
	`};`

6.16.6. DashboardBackend###0260-TicketCalendar

Description:	Parameters for the dashboard backend of the ticket calendar of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DashboardBackend'}->{'0260-TicketCalendar'} =  {`
	`  'Block' => 'ContentSmall',`
	`  'CacheTTL' => '2',`
	`  'Default' => '1',`
	`  'Group' => '',`
	`  'Limit' => '6',`
	`  'Module' => 'Kernel::Output::HTML::DashboardCalendar',`
	`  'OwnerOnly' => '',`
	`  'Permission' => 'rw',`
	`  'Title' => 'Upcoming Events'`
	`};`

6.16.7. DashboardBackend###0270-TicketQueueOverview

Description:	Parameters for the dashboard backend of the queue overview widget of the agent interface. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "QueuePermissionGroup" is not mandatory, queues are only listed if they belong to this permission group if you enable it. "States" is a list of states, the key is the sort order of the state in the widget. "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1

<table>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td><pre>
$Self->{'DashboardBackend'}->{'0270-TicketQueueOverview'} = {
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'Description' => 'Provides a matrix overview of the tickets per
state per queue.',
  'Group' => '',
  'Module' => 'Kernel::Output::HTML::DashboardTicketQueueOverview',
  'Permission' => 'rw',
  'QueuePermissionGroup' => 'users',
  'Sort' => 'SortBy=Age;OrderBy=Up',
  'States' => {
    '1' => 'new',
    '4' => 'open',
    '6' => 'pending reminder'
  },
  'Title' => 'Ticket Queue Overview'
};
</pre></td></tr>
</table>

6.16.8. DashboardBackend###0280-DashboardEventsTicketCalendar

<table>
<tr><td>Description:</td><td>Parameters for the dashboard backend of the ticket stats of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.</td></tr>
<tr><td>Group:</td><td>Ticket</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Agent::Dashboard</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td><pre>
$Self->{'DashboardBackend'}->{'0280-DashboardEventsTicketCalendar'} =
  {
  'Block' => 'ContentLarge',
  'CacheTTL' => '0',
  'Default' => '0',
  'Group' => '',
  'Module' => 'Kernel::Output::HTML::DashboardEventsTicketCalendar',
  'Title' => 'Events Ticket Calendar'
};
</pre></td></tr>
</table>

6.16.9. AgentCustomerInformationCenter::Backend###0100-CIC-TicketPendingReminder

<table>
<tr><td>Description:</td><td>Parameters for the dashboard backend of the ticket pending reminder overview of the agent interface . "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.</td></tr>
<tr><td>Group:</td><td>Ticket</td></tr>
</table>

SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:

```
 $Self->{'AgentCustomerInformationCenter::Backend'}->{'0100-CIC-
TicketPendingReminder'} =  {
  'Attributes' => 'TicketPendingTimeOlderMinutes=1;StateType=pending
 reminder;SortBy=PendingTime;OrderBy=Down;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All tickets with a reminder set where the reminder
 date has been reached',
  'Filter' => 'Locked',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
  'Permission' => 'ro',
  'Time' => 'UntilTime',
  'Title' => 'Reminder Tickets'
};
```

6.16.10. AgentCustomerInformationCenter::Backend###0110-CIC-TicketEscalation

Description:	Parameters for the dashboard backend of the ticket escalation overview of the agent interface . "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:

```
 $Self->{'AgentCustomerInformationCenter::Backend'}->{'0110-CIC-
TicketEscalation'} =  {
  'Attributes' =>
 'TicketEscalationTimeOlderMinutes=1;SortBy=EscalationTime;OrderBy=Down;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All escalated tickets',
  'Filter' => 'All',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
  'Permission' => 'ro',
  'Time' => 'EscalationTime',
  'Title' => 'Escalated Tickets'
};
```

6.16.11. AgentCustomerInformationCenter::Backend###0120-CIC-TicketNew

Description:	Parameters for the dashboard backend of the new tickets overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0

Config-Setting:

```
 $Self->{'AgentCustomerInformationCenter::Backend'}->{'0120-CIC-
TicketNew'} =  {
  'Attributes' => 'StateType=new;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All new tickets, these tickets have not been
 worked on yet',
  'Filter' => 'All',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
  'Permission' => 'ro',
  'Time' => 'Age',
  'Title' => 'New Tickets'
};
```

6.16.12. AgentCustomerInformationCenter::Backend###0130-CIC-TicketOpen

Description:	Parameters for the dashboard backend of the ticket pending reminder overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'AgentCustomerInformationCenter::Backend'}->{'0130-CIC-
TicketOpen'} =  {
  'Attributes' => 'StateType=open;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
```

<table>
<tr><td></td><td><pre>
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All open tickets, these tickets have already been
 worked on, but need a response',
  'Filter' => 'All',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric',
  'Permission' => 'ro',
  'Time' => 'Age',
  'Title' => 'Open Tickets / Need to be answered'
};
</pre></td></tr>
</table>

6.16.13. AgentCustomerInformationCenter::Backend###0500-CIC-CustomerIDStatus

<table>
<tr><td>Description:</td><td>Parameters for the dashboard backend of the customer id status widget of the agent interface . "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.</td></tr>
<tr><td>Group:</td><td>Ticket</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Agent::Dashboard</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'AgentCustomerInformationCenter::Backend'}->{'0500-CIC-
CustomerIDStatus'} = {
  'Attributes' => '',
  'Block' => 'ContentSmall',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'Description' => 'Company Status',
  'Group' => '',
  'Module' => 'Kernel::Output::HTML::DashboardCustomerIDStatus',
  'Permission' => 'ro',
  'Title' => 'Company Status'
};
</pre></td></tr>
</table>

6.17. Frontend::Agent::Dashboard::EventsTicketCalendar

6.17.1. DashboardEventsTicketCalendar###CalendarWidth

Description:	Defines the calendar width in percent. Default is 95%.
Group:	Ticket

SubGroup:	Frontend::Agent::Dashboard::EventsTicketCalendar
Valid:	1
Required:	1
Config-Setting:	`$Self->{'DashboardEventsTicketCalendar'}->{'CalendarWidth'} =  '95';`

6.17.2. DashboardEventsTicketCalendar###Queues

Description:	Defines queues that's tickets are used for displaying as calendar events.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard::EventsTicketCalendar
Valid:	1
Required:	1

Config-Setting:	`$Self->{'DashboardEventsTicketCalendar'}->{'Queues'} =  [`
	`  'Raw'`
	`];`

6.17.3. DashboardEventsTicketCalendar::DynamicFieldStartTime

Description:	Define dynamic field name for start time. This field has to be manually added to the system as Ticket: "Date / Time" and must be activated in ticket creation screens and/or in any other ticket action screens.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard::EventsTicketCalendar
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DashboardEventsTicketCalendar::DynamicFieldStartTime'} =`
	`'TicketCalendarStartTime';`

6.17.4. DashboardEventsTicketCalendar::DynamicFieldEndTime

Description:	Define dynamic field name for end time. This field has to be manually added to the system as Ticket: "Date / Time" and must be activated in ticket creation screens and/or in any other ticket action screens.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard::EventsTicketCalendar
Valid:	1
Required:	0

Config-Setting:	`$Self->{'DashboardEventsTicketCalendar::DynamicFieldEndTime'} =`
	`'TicketCalendarEndTime';`

6.17.5. DashboardEventsTicketCalendar::DynamicFieldsForEvents

Description:	Defines the dynamic fields that are used for displaying on calendar events.
Group:	Ticket

SubGroup:	Frontend::Agent::Dashboard::EventsTicketCalendar
Valid:	1
Required:	1

Config-Setting:	`$Self->{'DashboardEventsTicketCalendar::DynamicFieldsForEvents'} =  [`
	`  'TicketCalendarStartTime',`
	`  'TicketCalendarEndTime'`
	`];`

6.17.6. DashboardEventsTicketCalendar::TicketFieldsForEvents

Description:	Defines the ticket fields that are going to be displayed calendar events. The "Key" defines the field or ticket attribute and the "Content" defines the display name.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard::EventsTicketCalendar
Valid:	1
Required:	1

Config-Setting:	`$Self->{'DashboardEventsTicketCalendar::TicketFieldsForEvents'} =  {`
	`  'CustomerID' => 'Customer ID',`
	`  'CustomerUserID' => 'Customer user',`
	`  'Priority' => 'Priority',`
	`  'Queue' => 'Queue',`
	`  'SLA' => 'SLA',`
	`  'Service' => 'Service',`
	`  'State' => 'State',`
	`  'Title' => 'Title',`
	`  'Type' => 'Type'`
	`};`

6.18. Frontend::Agent::Dashboard::TicketFilters

6.18.1. OnlyValuesOnTicket

Description:	Defines if the list for filters should be retrieve just from current tickets in system. Just for clarification, Customers list will always came from system's tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard::TicketFilters
Valid:	1
Required:	1
Config-Setting:	`$Self->{'OnlyValuesOnTicket'} =  '1';`

6.19. Frontend::Agent::ModuleMetaHead

6.19.1. Frontend::HeaderMetaModule###2-TicketSearch

Description:	Module to generate html OpenSearch profile for short ticket search in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleMetaHead
Valid:	1

Required:	0

Config-Setting:	$Self->{'Frontend::HeaderMetaModule'}->{'2-TicketSearch'} = {
	  'Action' => 'AgentTicketSearch',
	  'Module' => 'Kernel::Output::HTML::HeaderMetaTicketSearch'
	};

6.20. Frontend::Agent::ModuleNotify

6.20.1. Frontend::NotifyModule###5-Ticket::TicketEscalation

Description:	Module to show notifications and escalations (ShownMax: max. shown escalations, EscalationInMinutes: Show ticket which will escalation in, CacheTime: Cache of calculated escalations in seconds).
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0

Config-Setting:	$Self->{'Frontend::NotifyModule'}->{'5-Ticket::TicketEscalation'} =
	{
	  'CacheTime' => '40',
	  'EscalationInMinutes' => '120',
	  'Module' =>
	'Kernel::Output::HTML::NotificationAgentTicketEscalation',
	  'ShownMax' => '25'
	};

6.21. Frontend::Agent::ModuleRegistration

6.21.1. Frontend::Module###AgentTicketQueue

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentTicketQueue'} = {
	  'Description' => 'Overview of all open Tickets',
	  'Loader' => {
	    'CSS' => [
	      'Core.AgentTicketQueue.css',
	      'Core.AllocationList.css'
	    ],
	    'JavaScript' => [
	      'Core.UI.AllocationList.js',
	      'Core.Agent.TableFilters.js'
	    ]
	  },
	  'NavBar' => [
	    {
	      'AccessKey' => 'o',
	      'Block' => '',
	      'Description' => 'Overview of all open Tickets',
	      'Link' => 'Action=AgentTicketQueue',
	      'LinkOption' => '',
	      'Name' => 'Queue view',
	      'NavBar' => 'Ticket',

```
        'Prio' => '100',
        'Type' => ''
      },
      {
        'AccessKey' => 't',
        'Block' => 'ItemArea',
        'Description' => '',
        'Link' => 'Action=AgentTicketQueue',
        'LinkOption' => '',
        'Name' => 'Tickets',
        'NavBar' => 'Ticket',
        'Prio' => '200',
        'Type' => 'Menu'
      }
    ],
    'NavBarName' => 'Ticket',
    'Title' => 'QueueView'
};
```

6.21.2. Frontend::Module###AgentTicketService

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AgentTicketService'} =  {
  'Description' => 'Overview of all open Tickets',
  'Loader' => {
    'CSS' => [
      'Core.AgentTicketService.css',
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.UI.AllocationList.js',
      'Core.Agent.TableFilters.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'O',
      'Block' => '',
      'Description' => 'Overview of all open Tickets',
      'Link' => 'Action=AgentTicketService',
      'LinkOption' => '',
      'Name' => 'Service view',
      'NavBar' => 'Ticket',
      'Prio' => '105',
      'Type' => ''
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'ServiceView'
};
```

6.21.3. Frontend::Module###AgentTicketPhone

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration

Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AgentTicketPhone'} =  {
  'Description' => 'Create new phone ticket',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.CustomerSearch.js',
      'Core.Agent.TicketAction.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'n',
      'Block' => '',
      'Description' => 'Create new phone ticket (inbound)',
      'Link' => 'Action=AgentTicketPhone',
      'LinkOption' => '',
      'Name' => 'New phone ticket',
      'NavBar' => 'Ticket',
      'Prio' => '200',
      'Type' => ''
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'New phone ticket'
};
```

6.21.4. Frontend::Module###AgentTicketPhoneOutbound

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AgentTicketPhoneOutbound'} =  {
  'Description' => 'Phone Call',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketAction.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Phone-Ticket'
};
```

6.21.5. Frontend::Module###AgentTicketPhoneInbound

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Module'}->{'AgentTicketPhoneInbound'} =  {
  'Description' => 'Incoming Phone Call',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketAction.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Phone-Ticket'
};
```

6.21.6. Frontend::Module###AgentTicketEmail

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
 $Self->{'Frontend::Module'}->{'AgentTicketEmail'} =  {
  'Description' => 'Create new email ticket',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.CustomerSearch.js',
      'Core.Agent.TicketAction.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'm',
      'Block' => '',
      'Description' => 'Create new email ticket and send this out
 (outbound)',
      'Link' => 'Action=AgentTicketEmail',
      'LinkOption' => '',
      'Name' => 'New email ticket',
      'NavBar' => 'Ticket',
      'Prio' => '210',
      'Type' => ''
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'New email ticket'
};
```

6.21.7. Frontend::Module###AgentTicketSearch

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
 $Self->{'Frontend::Module'}->{'AgentTicketSearch'} =  {
  'Description' => 'Search Ticket',
  'Loader' => {
    'JavaScript' => [
      'Core.UI.AllocationList.js',
      'Core.Agent.TableFilters.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 's',
      'Block' => '',
      'Description' => 'Search Tickets',
      'Link' => 'Action=AgentTicketSearch',
      'LinkOption' => 'onclick="window.setTimeout(function()
{Core.Agent.Search.OpenSearchDialog(\'AgentTicketSearch\');}, 0);
 return false;"',
      'Name' => 'Search',
      'NavBar' => 'Ticket',
      'Prio' => '300',
      'Type' => ''
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'Search'
};
```

6.21.8. Frontend::Module###AgentTicketLockedView

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
 $Self->{'Frontend::Module'}->{'AgentTicketLockedView'} =  {
  'Description' => 'Locked Tickets',
  'Loader' => {
    'CSS' => [
      'Core.AgentTicketQueue.css',
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.UI.AllocationList.js',
      'Core.Agent.TableFilters.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Locked Tickets'
};
```

6.21.9. Frontend::Module###AgentTicketResponsibleView

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:

```
 $Self->{'Frontend::Module'}->{'AgentTicketResponsibleView'} =  {
  'Description' => 'Responsible Tickets',
  'Loader' => {
    'CSS' => [
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.UI.AllocationList.js',
      'Core.Agent.TableFilters.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Responsible Tickets'
};
```

6.21.10. Frontend::Module###AgentTicketWatchView

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentTicketWatchView'} = {
	  'Description' => 'Watched Tickets',
	  'Loader' => {
	    'CSS' => [
	      'Core.AgentTicketQueue.css',
	      'Core.AllocationList.css'
	    ],
	    'JavaScript' => [
	      'Core.UI.AllocationList.js',
	      'Core.Agent.TableFilters.js'
	    ]
	  },
	  'NavBarName' => 'Ticket',
	  'Title' => 'Watched Tickets'
	};

6.21.11. Frontend::Module###AgentCustomerSearch

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentCustomerSearch'} = {
	  'Description' => 'AgentCustomerSearch',
	  'NavBarName' => 'Ticket',
	  'Title' => 'AgentCustomerSearch'
	};

6.21.12. Frontend::Module###AgentUserSearch

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentUserSearch'} = {
	  'Description' => 'AgentUserSearch',
	  'NavBarName' => 'Ticket',
	  'Title' => 'AgentUserSearch'
	};

6.21.13. Frontend::Module###AgentTicketStatusView

Description:	Frontend module registration for the agent interface.
Group:	Ticket

SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AgentTicketStatusView'} =  {
  'Description' => 'Overview of all open tickets',
  'Loader' => {
    'CSS' => [
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.UI.AllocationList.js',
      'Core.Agent.TableFilters.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'v',
      'Block' => '',
      'Description' => 'Overview of all open Tickets.',
      'Link' => 'Action=AgentTicketStatusView',
      'LinkOption' => '',
      'Name' => 'Status view',
      'NavBar' => 'Ticket',
      'Prio' => '110',
      'Type' => ''
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'Status view'
};
```

6.21.14. Frontend::Module###AgentTicketEscalationView

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AgentTicketEscalationView'} =  {
  'Description' => 'Overview of all escalated tickets',
  'Loader' => {
    'CSS' => [
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.UI.AllocationList.js',
      'Core.Agent.TableFilters.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'e',
      'Block' => '',
      'Description' => 'Overview Escalated Tickets',
      'Link' => 'Action=AgentTicketEscalationView',
      'LinkOption' => '',
      'Name' => 'Escalation view',
      'NavBar' => 'Ticket',
      'Prio' => '120',
      'Type' => ''
```

	`    }`
	`  ],`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'Escalation view'`
	`};`

6.21.15. Frontend::Module###AgentTicketZoom

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketZoom'} =  {`
	`  'Description' => 'Ticket Zoom',`
	`  'Loader' => {`
	`    'CSS' => [`
	`      'Core.Agent.TicketProcess.css'`
	`    ],`
	`    'JavaScript' => [`
	`      'thirdparty/jquery-tablesorter-2.0.5/jquery.tablesorter.js',`
	`      'Core.UI.Table.Sort.js',`
	`      'Core.Agent.TicketZoom.js'`
	`    ]`
	`  },`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'Zoom'`
	`};`

6.21.16. Frontend::Module###AgentTicketAttachment

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketAttachment'} =  {`
	`  'Description' => 'To download attachments',`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => ''`
	`};`

6.21.17. Frontend::Module###AgentTicketPlain

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketPlain'} =  {`
	`  'Description' => 'Ticket plain view of an email',`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'Plain'`
	`};`

6.21.18. Frontend::Module###AgentTicketNote

<table>
<tr><td>Description:</td><td>Frontend module registration for the agent interface.</td></tr>
<tr><td>Group:</td><td>Ticket</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Agent::ModuleRegistration</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'Frontend::Module'}->{'AgentTicketNote'} = {
  'Description' => 'Ticket Note',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketAction.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Note'
};
</pre></td></tr>
</table>

6.21.19. Frontend::Module###AgentTicketMerge

<table>
<tr><td>Description:</td><td>Frontend module registration for the agent interface.</td></tr>
<tr><td>Group:</td><td>Ticket</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Agent::ModuleRegistration</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'Frontend::Module'}->{'AgentTicketMerge'} = {
  'Description' => 'Ticket Merge',
  'NavBarName' => 'Ticket',
  'Title' => 'Merge'
};
</pre></td></tr>
</table>

6.21.20. Frontend::Module###AgentTicketPending

<table>
<tr><td>Description:</td><td>Frontend module registration for the agent interface.</td></tr>
<tr><td>Group:</td><td>Ticket</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Agent::ModuleRegistration</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'Frontend::Module'}->{'AgentTicketPending'} = {
  'Description' => 'Ticket Pending',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketAction.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Pending'
};
</pre></td></tr>
</table>

6.21.21. Frontend::Module###AgentTicketWatcher

Description:	Frontend module registration for the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentTicketWatcher'} = {
	  'Description' => 'A TicketWatcher Module',
	  'NavBarName' => 'Ticket-Watcher',
	  'Title' => 'Ticket-Watcher'
	};

6.21.22. Frontend::Module###AgentTicketPriority

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentTicketPriority'} = {
	  'Description' => 'Ticket Priority',
	  'Loader' => {
	    'JavaScript' => [
	      'Core.Agent.TicketAction.js'
	    ]
	  },
	  'NavBarName' => 'Ticket',
	  'Title' => 'Priority'
	};

6.21.23. Frontend::Module###AgentTicketLock

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentTicketLock'} = {
	  'Description' => 'Ticket Lock',
	  'NavBarName' => 'Ticket',
	  'Title' => 'Lock'
	};

6.21.24. Frontend::Module###AgentTicketMove

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentTicketMove'} = {
	  'Description' => 'Ticket Move',
	  'Loader' => {

	`    'JavaScript' => [`
	`      'Core.Agent.TicketAction.js'`
	`    ]`
	`  },`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'Move'`
	`};`

6.21.25. Frontend::Module###AgentTicketHistory

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketHistory'} =  {`
	`  'Description' => 'Ticket History',`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'History'`
	`};`

6.21.26. Frontend::Module###AgentTicketOwner

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketOwner'} =  {`
	`  'Description' => 'Ticket Owner',`
	`  'Loader' => {`
	`    'JavaScript' => [`
	`      'Core.Agent.TicketAction.js'`
	`    ]`
	`  },`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'Owner'`
	`};`

6.21.27. Frontend::Module###AgentTicketResponsible

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketResponsible'} =  {`
	`  'Description' => 'Ticket Responsible',`
	`  'Loader' => {`
	`    'JavaScript' => [`
	`      'Core.Agent.TicketAction.js'`
	`    ]`
	`  },`
	`  'NavBarName' => 'Ticket',`

	'Title' => 'Responsible' };

6.21.28. Frontend::Module###AgentTicketCompose

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentTicketCompose'} = {
	  'Description' => 'Ticket Compose email Answer',
	  'Loader' => {
	    'JavaScript' => [
	      'Core.Agent.CustomerSearch.js',
	      'Core.Agent.TicketAction.js'
	    ]
	  },
	  'NavBarName' => 'Ticket',
	  'Title' => 'Compose'
	};

6.21.29. Frontend::Module###AgentTicketBounce

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentTicketBounce'} = {
	  'Description' => 'Ticket Compose Bounce Email',
	  'NavBarName' => 'Ticket',
	  'Title' => 'Bounce'
	};

6.21.30. Frontend::Module###AgentTicketForward

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentTicketForward'} = {
	  'Description' => 'Ticket Forward Email',
	  'Loader' => {
	    'JavaScript' => [
	      'Core.Agent.CustomerSearch.js',
	      'Core.Agent.TicketAction.js'
	    ]
	  },
	  'NavBarName' => 'Ticket',
	  'Title' => 'Forward'
	};

6.21.31. Frontend::Module###AgentTicketCustomer

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AgentTicketCustomer'} =  {
  'Description' => 'Ticket Customer',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.CustomerSearch.js',
      'Core.Agent.TicketAction.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Customer'
};
```

6.21.32. Frontend::Module###AgentTicketClose

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AgentTicketClose'} =  {
  'Description' => 'Ticket Close',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketAction.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Close'
};
```

6.21.33. Frontend::Module###AgentTicketFreeText

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AgentTicketFreeText'} =  {
  'Description' => 'Ticket FreeText',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketAction.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Free Fields'
};
```

6.21.34. Frontend::Module###AgentTicketPrint

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentTicketPrint'} = {
	  'Description' => 'Ticket Print',
	  'NavBarName' => 'Ticket',
	  'Title' => 'Print'
	};

6.21.35. Frontend::Module###AgentTicketBulk

Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::Module'}->{'AgentTicketBulk'} = {
	  'Description' => 'Ticket bulk module',
	  'Loader' => {
	    'JavaScript' => [
	      'Core.Agent.TicketAction.js'
	    ]
	  },
	  'NavBarName' => 'Ticket',
	  'Title' => 'Bulk-Action'
	};

6.22. Frontend::Agent::Preferences

6.22.1. PreferencesGroups###NewTicketNotify

Description:	Parameters for the NewTicketNotify object in the preferences view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	$Self->{'PreferencesGroups'}->{'NewTicketNotify'} = {
	  'Active' => '1',
	  'Column' => 'Email Settings',
	  'Data' => {
	    '0' => 'No',
	    'MyQueues' => 'My Queues',
	    'MyQueuesAndMyServices' => 'My Queues and My Services',
	    'MyQueuesOrMyServices' => 'My Queues or My Services',
	    'MyServices' => 'My Services'
	  },
	  'DataSelected' => '0',
	  'Desc' => 'Send me a notification if there is a new ticket in my subscriptions.',
	  'Key' => 'Send new ticket notifications if subscribed to:',

```
  'Label' => 'New ticket notification',
  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
  'PrefKey' => 'UserSendNewTicketNotification',
  'Prio' => '1000'
};
```

6.22.2. PreferencesGroups###FollowUpNotify

Description:	Parameters for the FollowUpNotify object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'PreferencesGroups'}->{'FollowUpNotify'} =  {
  'Active' => '1',
  'Column' => 'Email Settings',
  'Data' => {
    '0' => 'No',
    'MyQueues' => 'My Queues',
    'MyQueuesAndMyServices' => 'My Queues and My Services',
    'MyQueuesOrMyServices' => 'My Queues or My Services',
    'MyServices' => 'My Services'
  },
  '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 subscriptions.',
  'Key' => 'Send ticket follow up notifications if subscribed to:',
  'Label' => 'Ticket follow up notification',
  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
  'PrefKey' => 'UserSendFollowUpNotification',
  'Prio' => '2000'
};
```

6.22.3. PreferencesGroups###LockTimeoutNotify

Description:	Parameters for the LockTimeoutNotify object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'PreferencesGroups'}->{'LockTimeoutNotify'} =  {
  'Active' => '1',
  'Column' => 'Email Settings',
  'Data' => {
    '0' => 'No',
    '1' => 'Yes'
  },
  'DataSelected' => '0',
  'Desc' => 'Send me a notification if a ticket is unlocked by the
 system.',
  'Key' => 'Send ticket lock timeout notifications',
  'Label' => 'Ticket lock timeout notification',
  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
  'PrefKey' => 'UserSendLockTimeoutNotification',
  'Prio' => '3000'
```

	};

6.22.4. PreferencesGroups###MoveNotify

Description:	Parameters for the MoveNotify object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	$Self->{'PreferencesGroups'}->{'MoveNotify'} = {
	  'Active' => '1',
	  'Column' => 'Email Settings',
	  'Data' => {
	    '0' => 'No',
	    '1' => 'Yes'
	  },
	  'DataSelected' => '0',
	  'Desc' => 'Send me a notification if a ticket is moved into one of "My Queues".',
	  'Key' => 'Send ticket move notifications',
	  'Label' => 'Ticket move notification',
	  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	  'PrefKey' => 'UserSendMoveNotification',
	  'Prio' => '4000'
	};

6.22.5. PreferencesGroups###ServiceUpdateNotify

Description:	Parameters for the ServiceUpdateNotify object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	$Self->{'PreferencesGroups'}->{'ServiceUpdateNotify'} = {
	  'Active' => '1',
	  'Column' => 'Email Settings',
	  'Data' => {
	    '0' => 'No',
	    '1' => 'Yes'
	  },
	  'DataSelected' => '0',
	  'Desc' => 'Send me a notification if the service of a ticket is changed to a service in "My Services" and the ticket is in a queue where I have read permissions.',
	  'Key' => 'Send service update notifications',
	  'Label' => 'Service update notification',
	  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	  'PrefKey' => 'UserSendServiceUpdateNotification',
	  'Prio' => '4000'
	};

6.22.6. PreferencesGroups###WatcherNotify

Description:	Parameters for the WatcherNotify object in the preference view of the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'PreferencesGroups'}->{'WatcherNotify'} =  {
  'Active' => '1',
  'Column' => 'Email Settings',
  'Data' => {
    '0' => 'No',
    '1' => 'Yes'
  },
  'DataSelected' => '0',
  'Desc' => 'Send me the same notifications for my watched tickets
that the ticket owners will get.',
  'Key' => 'Send ticket watch notifications',
  'Label' => 'Ticket watch notification',
  'Module' => 'Kernel::Output::HTML::PreferencesTicketWatcher',
  'PrefKey' => 'UserSendWatcherNotification',
  'Prio' => '5000'
};
```

6.22.7. PreferencesGroups###CustomQueue

Description:	Parameters for the CustomQueue object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'PreferencesGroups'}->{'CustomQueue'} =  {
  'Active' => '1',
  'Column' => 'Other Settings',
  'Desc' => 'Your queue selection of your favorite queues. You also
get notified about those queues via email if enabled.',
  'Key' => 'My Queues',
  'Label' => 'My Queues',
  'Module' => 'Kernel::Output::HTML::PreferencesCustomQueue',
  'Permission' => 'ro',
  'Prio' => '1000'
};
```

6.22.8. PreferencesGroups###CustomService

Description:	Parameters for the CustomService object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'PreferencesGroups'}->{'CustomService'} =  {
  'Active' => '1',
  'Column' => 'Other Settings',
  'Desc' => 'Your service selection of your favorite services. You
also get notified about those services via email if enabled.',
  'Key' => 'My Services',
```

	'Label' => 'My Services', 'Module' => 'Kernel::Output::HTML::PreferencesCustomService', 'Prio' => '1000' };

6.22.9. PreferencesGroups###RefreshTime

Description:	Parameters for the RefreshTime object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	$Self->{'PreferencesGroups'}->{'RefreshTime'} = {
	  'Active' => '1',
	  'Column' => 'Other Settings',
	  'Data' => {
	    '0' => 'off',
	    '10' => '10 minutes',
	    '15' => '15 minutes',
	    '2' => ' 2 minutes',
	    '5' => ' 5 minutes',
	    '7' => ' 7 minutes'
	  },
	  'DataSelected' => '0',
	  'Desc' => 'If enabled, the different overviews (Dashboard, LockedView, QueueView) will automatically refresh after the specified time.',
	  'Key' => 'Refresh Overviews after',
	  'Label' => 'Overview Refresh Time',
	  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	  'PrefKey' => 'UserRefreshTime',
	  'Prio' => '2000'
	};

6.22.10. PreferencesGroups###TicketOverviewSmallPageShown

Description:	Parameters for the pages (in which the tickets are shown) of the small ticket overview.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	$Self->{'PreferencesGroups'}->{'TicketOverviewSmallPageShown'} = {
	  'Active' => '0',
	  'Column' => 'Other Settings',
	  'Data' => {
	    '10' => '10',
	    '15' => '15',
	    '20' => '20',
	    '25' => '25',
	    '30' => '30',
	    '35' => '35'
	  },
	  'DataSelected' => '25',
	  'Key' => 'Ticket limit per page for Ticket Overview "Small"',
	  'Label' => 'Ticket Overview "Small" Limit',
	  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',

	'PrefKey' => 'UserTicketOverviewSmallPageShown', 'Prio' => '8000' };

6.22.11. PreferencesGroups###TicketOverviewFilterSettings

Description:	Parameters for .
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	$Self->{'PreferencesGroups'}->{'TicketOverviewFilterSettings'} = {
	'Active' => '0',
	'Column' => 'Other Settings',
	'Key' => 'Column ticket filters for Ticket Overviews type "Small".',
	'Label' => 'Enabled filters.',
	'Module' => 'Kernel::Output::HTML::PreferencesColumnFilters',
	'PrefKey' => 'UserFilterColumnsEnabled',
	'Prio' => '8100'
	};

6.22.12. PreferencesGroups###TicketOverviewMediumPageShown

Description:	Parameters for the pages (in which the tickets are shown) of the medium ticket overview.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	$Self->{'PreferencesGroups'}->{'TicketOverviewMediumPageShown'} = {
	'Active' => '0',
	'Column' => 'Other Settings',
	'Data' => {
	'10' => '10',
	'15' => '15',
	'20' => '20',
	'25' => '25',
	'30' => '30',
	'35' => '35'
	},
	'DataSelected' => '20',
	'Key' => 'Ticket limit per page for Ticket Overview "Medium"',
	'Label' => 'Ticket Overview "Medium" Limit',
	'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	'PrefKey' => 'UserTicketOverviewMediumPageShown',
	'Prio' => '8100'
	};

6.22.13. PreferencesGroups###TicketOverviewPreviewPageShown

Description:	Parameters for the pages (in which the tickets are shown) of the ticket preview overview.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences

Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'TicketOverviewPreviewPageShown'} =  {`
	`  'Active' => '0',`
	`  'Column' => 'Other Settings',`
	`  'Data' => {`
	`    '10' => '10',`
	`    '15' => '15',`
	`    '20' => '20',`
	`    '25' => '25',`
	`    '30' => '30',`
	`    '35' => '35'`
	`  },`
	`  'DataSelected' => '15',`
	`  'Key' => 'Ticket limit per page for Ticket Overview "Preview"',`
	`  'Label' => 'Ticket Overview "Preview" Limit',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',`
	`  'PrefKey' => 'UserTicketOverviewPreviewPageShown',`
	`  'Prio' => '8200'`
	`};`

6.22.14. PreferencesGroups###CreateNextMask

Description:	Parameters for the CreateNextMask object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'CreateNextMask'} =  {`
	`  'Active' => '1',`
	`  'Column' => 'Other Settings',`
	`  'Data' => {`
	`    '' => 'CreateTicket',`
	`    'AgentTicketZoom' => 'TicketZoom'`
	`  },`
	`  'DataSelected' => '',`
	`  'Key' => 'Show this screen after I created a new ticket',`
	`  'Label' => 'Screen after new ticket',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',`
	`  'PrefKey' => 'UserCreateNextMask',`
	`  'Prio' => '3000'`
	`};`

6.23. Frontend::Agent::SearchRouter

6.23.1. Frontend::Search###AgentCustomerInformationCenter

Description:	Search backend router.
Group:	Ticket
SubGroup:	Frontend::Agent::SearchRouter
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Search'}->{'AgentCustomerInformationCenter'} =  {`
	`  '^AgentCustomerInformationCenter' =>`
	`'Action=AgentCustomerInformationCenterSearch'`

	```
	};
	```

6.23.2. Frontend::Search###Ticket

Description:	Search backend router.
Group:	Ticket
SubGroup:	Frontend::Agent::SearchRouter
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Search'}->{'Ticket'} =  {`
	`  '^AgentTicket' => 'Action=AgentTicketSearch;Subaction=AJAX'`
	`};`

6.24. Frontend::Agent::Ticket::ArticleAttachmentModule

6.24.1. Ticket::Frontend::ArticleAttachmentModule###1-Download

Description:	Shows a link to download article attachments in the zoom view of the article in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleAttachmentModule
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::ArticleAttachmentModule'}->{'1-Download'}`
	` =  {`
	`  'Module' => 'Kernel::Output::HTML::ArticleAttachmentDownload'`
	`};`

6.24.2. Ticket::Frontend::ArticleAttachmentModule###2-HTML-Viewer

Description:	Shows a link to access article attachments via a html online viewer in the zoom view of the article in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleAttachmentModule
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::ArticleAttachmentModule'}->{'2-HTML-`
	`Viewer'} =  {`
	`  'Module' => 'Kernel::Output::HTML::ArticleAttachmentHTMLViewer'`
	`};`

6.25. Frontend::Agent::Ticket::ArticleComposeModule

6.25.1. Ticket::Frontend::ArticleComposeModule###1-SignEmail

Description:	Module to compose signed messages (PGP or S/MIME).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleComposeModule
Valid:	1

Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::ArticleComposeModule'}->{'1-SignEmail'} =`
	`{`
	`'Module' => 'Kernel::Output::HTML::ArticleComposeSign'`
	`};`

6.25.2. Ticket::Frontend::ArticleComposeModule###2-CryptEmail

Description:	Module to crypt composed messages (PGP or S/MIME).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleComposeModule
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::ArticleComposeModule'}->{'2-CryptEmail'} =`
	`{`
	`'Module' => 'Kernel::Output::HTML::ArticleComposeCrypt'`
	`};`

6.26. Frontend::Agent::Ticket::ArticleViewModule

6.26.1. Ticket::Frontend::ArticleViewModule###1-PGP

Description:	Agent interface article notification module to check PGP.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModule
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::ArticleViewModule'}->{'1-PGP'} =  {`
	`'Module' => 'Kernel::Output::HTML::ArticleCheckPGP'`
	`};`

6.26.2. Ticket::Frontend::ArticleViewModule###1-SMIME

Description:	Agent interface module to check incoming emails in the Ticket-Zoom-View if the S/MIME-key is available and true.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModule
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::ArticleViewModule'}->{'1-SMIME'} =  {`
	`'Module' => 'Kernel::Output::HTML::ArticleCheckSMIME'`
	`};`

6.27. Frontend::Agent::Ticket::ArticleViewModulePre

6.27.1. Ticket::Frontend::ArticlePreViewModule###1-PGP

Description:	Agent interface article notification module to check PGP.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModulePre

Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::ArticlePreViewModule'}->{'1-PGP'} =  {`
	`  'Module' => 'Kernel::Output::HTML::ArticleCheckPGP'`
	`};`

6.27.2. Ticket::Frontend::ArticlePreViewModule###1-SMIME

Description:	Agent interface article notification module to check S/MIME.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModulePre
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::ArticlePreViewModule'}->{'1-SMIME'} =  {`
	`  'Module' => 'Kernel::Output::HTML::ArticleCheckSMIME'`
	`};`

6.28. Frontend::Agent::Ticket::MenuModule

6.28.1. Ticket::Frontend::MenuModule###000-Back

Description:	Shows a link in the menu to go back in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'000-Back'} =  {`
	`  'Action' => '',`
	`  'Description' => 'Back',`
	`  'Link' => '[% Env("LastScreenOverview") %];TicketID=[% Data.TicketID | html %]',`
	`  'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`  'Name' => 'Back',`
	`  'PopupType' => '',`
	`  'Target' => ''`
	`};`

6.28.2. Ticket::Frontend::MenuModule###100-Lock

Description:	Shows a link in the menu to lock/unlock tickets in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'100-Lock'} =  {`
	`  'Action' => 'AgentTicketLock',`
	`  'Module' => 'Kernel::Output::HTML::TicketMenuLock',`
	`  'Name' => 'Lock',`
	`  'Target' => ''`
	`};`

6.28.3. Ticket::Frontend::MenuModule###200-History

Description:	Shows a link in the menu to access the history of a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'200-History'} =  {
  'Action' => 'AgentTicketHistory',
  'Description' => 'Show the ticket history',
  'Link' => 'Action=AgentTicketHistory;TicketID=[% Data.TicketID |
 html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
  'Name' => 'History',
  'PopupType' => 'TicketHistory',
  'Target' => ''
};
```

6.28.4. Ticket::Frontend::MenuModule###210-Print

Description:	Shows a link in the menu to print a ticket or an article in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'210-Print'} =  {
  'Action' => 'AgentTicketPrint',
  'Description' => 'Print this ticket',
  'Link' => 'Action=AgentTicketPrint;TicketID=[% Data.TicketID | html
 %]',
  'LinkParam' => 'target="print"',
  'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
  'Name' => 'Print',
  'PopupType' => 'TicketAction',
  'Target' => ''
};
```

6.28.5. Ticket::Frontend::MenuModule###300-Priority

Description:	Shows a link in the menu to see the priority of a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'300-Priority'} =  {
  'Action' => 'AgentTicketPriority',
  'Description' => 'Change the ticket priority',
  'Link' => 'Action=AgentTicketPriority;TicketID=[% Data.TicketID |
 html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
  'Name' => 'Priority',
```

	'PopupType' => 'TicketAction', 'Target' => '' };

6.28.6. Ticket::Frontend::MenuModule###310-FreeText

Description:	Shows a link in the menu to add a free text field in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	$Self->{'Ticket::Frontend::MenuModule'}->{'310-FreeText'} = {
	'Action' => 'AgentTicketFreeText',
	'Description' => 'Change the free fields for this ticket',
	'Link' => 'Action=AgentTicketFreeText;TicketID=[% Data.TicketID | html %]',
	'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
	'Name' => 'Free Fields',
	'PopupType' => 'TicketAction',
	'Target' => ''
	};

6.28.7. Ticket::Frontend::MenuModule###320-Link

Description:	Shows a link in the menu that allows linking a ticket with another object in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	$Self->{'Ticket::Frontend::MenuModule'}->{'320-Link'} = {
	'Action' => 'AgentLinkObject',
	'Description' => 'Link this ticket to other objects',
	'Link' => 'Action=AgentLinkObject;SourceObject=Ticket;SourceKey=[% Data.TicketID | html %]',
	'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
	'Name' => 'Link',
	'PopupType' => 'TicketAction',
	'Target' => ''
	};

6.28.8. Ticket::Frontend::MenuModule###400-Owner

Description:	Shows a link in the menu to see the owner of a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::MenuModule'}->{'400-Owner'} = { 'Action' => 'AgentTicketOwner', 'Description' => 'Change the owner for this ticket',

	'Link' => 'Action=AgentTicketOwner;TicketID=[% Data.TicketID | html %]', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Owner', 'PopupType' => 'TicketAction', 'Target' => '' };

6.28.9. Ticket::Frontend::MenuModule###410-Responsible

Description:	Shows a link in the menu to see the responsible agent of a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	$Self->{'Ticket::Frontend::MenuModule'}->{'410-Responsible'} = {
	'Action' => 'AgentTicketResponsible',
	'Description' => 'Change the responsible person for this ticket',
	'Link' => 'Action=AgentTicketResponsible;TicketID=[% Data.TicketID | html %]',
	'Module' => 'Kernel::Output::HTML::TicketMenuResponsible',
	'Name' => 'Responsible',
	'PopupType' => 'TicketAction',
	'Target' => ''
	};

6.28.10. Ticket::Frontend::MenuModule###420-Customer

Description:	Shows a link in the menu to see the customer who requested the ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	$Self->{'Ticket::Frontend::MenuModule'}->{'420-Customer'} = {
	'Action' => 'AgentTicketCustomer',
	'Description' => 'Change the customer for this ticket',
	'Link' => 'Action=AgentTicketCustomer;TicketID=[% Data.TicketID | html %]',
	'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
	'Name' => 'Customer',
	'PopupType' => 'TicketAction',
	'Target' => ''
	};

6.28.11. Ticket::Frontend::MenuModule###420-Note

Description:	Shows a link in the menu to add a note in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'420-Note'} =  {`
	`  'Action' => 'AgentTicketNote',`
	`  'Description' => 'Add a note to this ticket',`
	`  'Link' => 'Action=AgentTicketNote;TicketID=[% Data.TicketID | html`
	` %]',`
	`  'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`  'Name' => 'Note',`
	`  'PopupType' => 'TicketAction',`
	`  'Target' => ''`
	`};`

6.28.12. Ticket::Frontend::MenuModule###425-Phone Call Outbound

Description:	Shows a link in the menu to add a note in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'425-Phone Call Outbound'}`
	` =  {`
	`  'Action' => 'AgentTicketPhoneOutbound',`
	`  'Description' => 'Phone Call Outbound',`
	`  'Link' => 'Action=AgentTicketPhoneOutbound;TicketID=[% Data.TicketID`
	` | html %]',`
	`  'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`  'Name' => 'Phone Call Outbound',`
	`  'PopupType' => 'TicketAction',`
	`  'Target' => ''`
	`};`

6.28.13. Ticket::Frontend::MenuModule###426-Phone Call Inbound

Description:	Shows a link in the menu to add a note in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'426-Phone Call Inbound'} =`
	`  {`
	`  'Action' => 'AgentTicketPhoneInbound',`
	`  'Description' => 'Phone Call Inbound',`
	`  'Link' => 'Action=AgentTicketPhoneInbound;TicketID=[% Data.TicketID`
	` | html %]',`
	`  'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`  'Name' => 'Phone Call Inbound',`
	`  'PopupType' => 'TicketAction',`
	`  'Target' => ''`
	`};`

6.28.14. Ticket::Frontend::MenuModule###430-Merge

Description:	Shows a link in the menu that allows merging tickets in the ticket zoom view of the agent interface.
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'430-Merge'} =  {`
	`  'Action' => 'AgentTicketMerge',`
	`  'Description' => 'Merge into a different ticket',`
	`  'Link' => 'Action=AgentTicketMerge;TicketID=[% Data.TicketID | html %]',`
	`  'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`  'Name' => 'Merge',`
	`  'PopupType' => 'TicketAction',`
	`  'Target' => ''`
	`};`

6.28.15. Ticket::Frontend::MenuModule###440-Pending

Description:	Shows a link in the menu to set a ticket as pending in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'440-Pending'} =  {`
	`  'Action' => 'AgentTicketPending',`
	`  'Description' => 'Set this ticket to pending',`
	`  'Link' => 'Action=AgentTicketPending;TicketID=[% Data.TicketID | html %]',`
	`  'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`  'Name' => 'Pending',`
	`  'PopupType' => 'TicketAction',`
	`  'Target' => ''`
	`};`

6.28.16. Ticket::Frontend::MenuModule###448-Watch

Description:	Shows a link in the menu for subscribing / unsubscribing from a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'448-Watch'} =  {`
	`  'Action' => 'AgentTicketWatcher',`
	`  'Module' => 'Kernel::Output::HTML::TicketMenuTicketWatcher',`
	`  'Name' => 'Watch',`
	`  'Target' => ''`
	`};`

6.28.17. Ticket::Frontend::MenuModule###450-Close

Description:	Shows a link in the menu to close a ticket in the ticket zoom view of the agent interface.
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'450-Close'} =  {`
	`  'Action' => 'AgentTicketClose',`
	`  'Description' => 'Close this ticket',`
	`  'Link' => 'Action=AgentTicketClose;TicketID=[% Data.TicketID | html`
	`%]',`
	`  'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`  'Name' => 'Close',`
	`  'PopupType' => 'TicketAction',`
	`  'Target' => ''`
	`};`

6.28.18. Ticket::Frontend::MenuModule###460-Delete

Description:	Shows a link in the menu to delete a ticket in the ticket zoom view of the agent interface. Additional access control to show or not show this link can be done by using Key "Group" and Content like "rw:group1;move_into:group2".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	0
Required:	0

Config-Setting:	`$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',`
	`  'PopupType' => '',`
	`  'Target' => ''`
	`};`

6.28.19. Ticket::Frontend::MenuModule###470-Spam

Description:	Shows a link to set a ticket as spam in the ticket zoom view of the agent interface. Additional access control to show or not show this link can be done by using Key "Group" and Content like "rw:group1;move_into:group2".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	0
Required:	0

Config-Setting:	`$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',`
	`  'PopupType' => '',`
	`  'Target' => ''`

	};

6.29. Frontend::Agent::Ticket::MenuModulePre

6.29.1. Ticket::Frontend::PreMenuModule###100-Lock

Description:	Shows a link in the menu to lock / unlock a ticket in the ticket overviews of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Config-Setting:	$Self->{'Ticket::Frontend::PreMenuModule'}->{'100-Lock'} = {
	 'Action' => 'AgentTicketLock',
	 'Module' => 'Kernel::Output::HTML::TicketMenuLock',
	 'Name' => 'Lock',
	 'PopupType' => '',
	 'Target' => ''
	};

6.29.2. Ticket::Frontend::PreMenuModule###200-Zoom

Description:	Shows a link in the menu to zoom a ticket in the ticket overviews of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Config-Setting:	$Self->{'Ticket::Frontend::PreMenuModule'}->{'200-Zoom'} = {
	 'Action' => 'AgentTicketZoom',
	 'Description' => 'Look into a ticket!',
	 'Link' => 'Action=AgentTicketZoom;TicketID=[% Data.TicketID | html %]',
	 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
	 'Name' => 'Zoom',
	 'PopupType' => '',
	 'Target' => ''
	};

6.29.3. Ticket::Frontend::PreMenuModule###210-History

Description:	Shows a link in the menu to see the history of a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::PreMenuModule'}->{'210-History'} = { 'Action' => 'AgentTicketHistory', 'Description' => 'Show the ticket history', 'Link' => 'Action=AgentTicketHistory;TicketID=[% Data.TicketID | html %]', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'History',

	'PopupType' => 'TicketHistory', 'Target' => '' };

6.29.4. Ticket::Frontend::PreMenuModule###300-Priority

Description:	Shows a link in the menu to set the priority of a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Config-Setting:	$Self->{'Ticket::Frontend::PreMenuModule'}->{'300-Priority'} = {
	 'Action' => 'AgentTicketPriority',
	 'Description' => 'Change the priority for this ticket',
	 'Link' => 'Action=AgentTicketPriority;TicketID=[% Data.TicketID | html %]',
	 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
	 'Name' => 'Priority',
	 'PopupType' => 'TicketAction',
	 'Target' => ''
	};

6.29.5. Ticket::Frontend::PreMenuModule###420-Note

Description:	Shows a link in the menu to add a note to a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Config-Setting:	$Self->{'Ticket::Frontend::PreMenuModule'}->{'420-Note'} = {
	 'Action' => 'AgentTicketNote',
	 'Description' => 'Add a note to this ticket',
	 'Link' => 'Action=AgentTicketNote;TicketID=[% Data.TicketID | html %]',
	 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
	 'Name' => 'Note',
	 'PopupType' => 'TicketAction',
	 'Target' => ''
	};

6.29.6. Ticket::Frontend::PreMenuModule###440-Close

Description:	Shows a link in the menu to close a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::PreMenuModule'}->{'440-Close'} = { 'Action' => 'AgentTicketClose', 'Description' => 'Close this ticket',

	'Link' => 'Action=AgentTicketClose;TicketID=[% Data.TicketID | html %]', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Close', 'PopupType' => 'TicketAction', 'Target' => '' };

6.29.7. Ticket::Frontend::PreMenuModule###445-Move

Description:	Shows a link in the menu to move a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Config-Setting:	$Self->{'Ticket::Frontend::PreMenuModule'}->{'445-Move'} = {
	'Action' => 'AgentTicketMove',
	'Description' => 'Change queue!',
	'Module' => 'Kernel::Output::HTML::TicketMenuMove',
	'Name' => 'Move'
	};

6.29.8. Ticket::Frontend::PreMenuModule###450-Delete

Description:	Shows a link in the menu to delete a ticket in every ticket overview of the agent interface. Additional access control to show or not show this link can be done by using Key "Group" and Content like "rw:group1;move_into:group2".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	0
Required:	0

Config-Setting:	$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',
	'PopupType' => '',
	'Target' => ''
	};

6.29.9. Ticket::Frontend::PreMenuModule###460-Spam

Description:	Shows a link in the menu to set a ticket as spam in every ticket overview of the agent interface. Additional access control to show or not show this link can be done by using Key "Group" and Content like "rw:group1;move_into:group2".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	0

Required:	0

Config-Setting:	`$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',`
	`  'PopupType' => '',`
	`  'Target' => ''`
	`};`

6.30. Frontend::Agent::Ticket::OverviewMenuModule

6.30.1. Ticket::Frontend::OverviewMenuModule###001-Sort

Description:	Shows a select of ticket attributes to order the queue view ticket list. The possible selections can be configured via 'TicketOverviewMenuSort###SortAttributes'.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::OverviewMenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::OverviewMenuModule'}->{'001-Sort'} =  {`
	`  'Module' => 'Kernel::Output::HTML::TicketOverviewMenuSort'`
	`};`

6.30.2. TicketOverviewMenuSort###SortAttributes

Description:	Defines from which ticket attributes the agent can select the result order.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::OverviewMenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'TicketOverviewMenuSort'}->{'SortAttributes'} =  {`
	`  'Age' => '1',`
	`  'Title' => '1'`
	`};`

6.31. Frontend::Agent::Ticket::ViewBounce

6.31.1. Ticket::Frontend::AgentTicketBounce###Permission

Description:	Required permissions to use the ticket bounce screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'Permission'} =`
	`'bounce';`

6.31.2. Ticket::Frontend::AgentTicketBounce###RequiredLock

Description:	Defines if a ticket lock is required in the ticket bounce screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'RequiredLock'} = '1';`

6.31.3. Ticket::Frontend::AgentTicketBounce###StateDefault

Description:	Defines the default next state of a ticket after being bounced, in the ticket bounce screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'StateDefault'} = 'closed successful';`

6.31.4. Ticket::Frontend::AgentTicketBounce###StateType

Description:	Defines the next state of a ticket after being bounced, in the ticket bounce screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'StateType'} =  [`
	`  'open',`
	`  'closed'`
	`];`

6.31.5. Ticket::Frontend::BounceText

Description:	Defines the default ticket bounced notification for customer/sender in the ticket bounce screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::BounceText'} =  'Your email with ticket number "<OTRS_TICKET>" is bounced to "<OTRS_BOUNCE_TO>". Contact this address for further information.';`

6.32. Frontend::Agent::Ticket::ViewBulk

6.32.1. Ticket::Frontend::AgentTicketBulk###RequiredLock

Description:	Automatically lock and set owner to current Agent after selecting for an Bulk Action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'RequiredLock'} = '1';`

6.32.2. Ticket::Frontend::AgentTicketBulk###TicketType

Description:	Sets the ticket type in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'TicketType'} =  '1';`

6.32.3. Ticket::Frontend::AgentTicketBulk###Owner

Description:	Sets the ticket owner in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Owner'} =  '1';`

6.32.4. Ticket::Frontend::AgentTicketBulk###Responsible

Description:	Sets the responsible agent of the ticket in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Responsible'} =  '1';`

6.32.5. Ticket::Frontend::AgentTicketBulk###State

Description:	If a note is added by an agent, sets the state of a ticket in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk

Valid:	1
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'State'} = '1';

6.32.6. Ticket::Frontend::AgentTicketBulk###StateType

Description:	Defines the next state of a ticket after adding a note, in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	1

Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'StateType'} = [
	  'open',
	  'closed',
	  'pending reminder',
	  'pending auto'
	];

6.32.7. Ticket::Frontend::AgentTicketBulk###StateDefault

Description:	Defines the default next state of a ticket after adding a note, in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	0
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'StateDefault'} = 'open';

6.32.8. Ticket::Frontend::AgentTicketBulk###Priority

Description:	Shows the ticket priority options in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Priority'} = '1';

6.32.9. Ticket::Frontend::AgentTicketBulk###PriorityDefault

Description:	Defines the default ticket priority in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'PriorityDefault'} = '3 normal';`

6.32.10. Ticket::Frontend::AgentTicketBulk###ArticleTypeDefault

Description:	Defines the default type of the note in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'ArticleTypeDefault'} =  'note-internal';`

6.32.11. Ticket::Frontend::AgentTicketBulk###ArticleTypes

Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'ArticleTypes'} =  {`
	`  'note-external' => '1',`
	`  'note-internal' => '1',`
	`  'note-report' => '0'`
	`};`

6.33. Frontend::Agent::Ticket::ViewClose

6.33.1. Ticket::Frontend::AgentTicketClose###Permission

Description:	Required permissions to use the close ticket screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Permission'} = 'close';`

6.33.2. Ticket::Frontend::AgentTicketClose###RequiredLock

Description:	Defines if a ticket lock is required in the close ticket screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'RequiredLock'} = '1';`

6.33.3. Ticket::Frontend::AgentTicketClose###TicketType

Description:	Sets the ticket type in the close ticket screen of the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'TicketType'} =  '0';`

6.33.4. Ticket::Frontend::AgentTicketClose###Service

Description:	Sets the service in the close ticket screen of the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Service'} =  '0';`

6.33.5. Ticket::Frontend::AgentTicketClose###ServiceMandatory

Description:	Sets if service must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ServiceMandatory'} =  '0';`

6.33.6. Ticket::Frontend::AgentTicketClose###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'SLAMandatory'} = '0';`

6.33.7. Ticket::Frontend::AgentTicketClose###Queue

Description:	Sets the queue in the ticket close screen of a zoomed ticket in the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Queue'} =  '0';`

6.33.8. Ticket::Frontend::AgentTicketClose###Owner

Description:	Sets the ticket owner in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Owner'} =  '0';`

6.33.9. Ticket::Frontend::AgentTicketClose###OwnerMandatory

Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'OwnerMandatory'} = '0';`

6.33.10. Ticket::Frontend::AgentTicketClose###Responsible

Description:	Sets the responsible agent of the ticket in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Responsible'} = '0';`

6.33.11. Ticket::Frontend::AgentTicketClose###State

Description:	If a note is added by an agent, sets the state of a ticket in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'State'} =  '1';`

6.33.12. Ticket::Frontend::AgentTicketClose###StateType

Description:	Defines the next state of a ticket after adding a note, in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'StateType'} =  [`
	`  'closed'`
	`];`

6.33.13. Ticket::Frontend::AgentTicketClose###StateDefault

Description:	Defines the default next state of a ticket after adding a note, in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'StateDefault'} =`
	`'closed successful';`

6.33.14. Ticket::Frontend::AgentTicketClose###Note

Description:	Allows adding notes in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Note'} =  '1';`

6.33.15. Ticket::Frontend::AgentTicketClose###NoteMandatory

Description:	Sets if note must be filled in by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'NoteMandatory'} =`
	`'1';`

6.33.16. Ticket::Frontend::AgentTicketClose###Subject

Description:	Sets the default subject for notes added in the close ticket screen of the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Subject'} = '[% Translate("Close") | html %]';`

6.33.17. Ticket::Frontend::AgentTicketClose###Body

Description:	Sets the default body text for notes added in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Body'} = '';`

6.33.18. Ticket::Frontend::AgentTicketClose###InvolvedAgent

Description:	Shows a list of all the involved agents on this ticket, in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'InvolvedAgent'} = '0';`

6.33.19. Ticket::Frontend::AgentTicketClose###InformAgent

Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'InformAgent'} = '0';`

6.33.20. Ticket::Frontend::AgentTicketClose###ArticleTypeDefault

Description:	Defines the default type of the note in the close ticket screen of the agent interface.
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ArticleTypeDefault'}`
	`=  'note-internal';`

6.33.21. Ticket::Frontend::AgentTicketClose###ArticleTypes

Description:	Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ArticleTypes'} =  {`
	`  'note-external' => '0',`
	`  'note-internal' => '1',`
	`  'note-report' => '0'`
	`};`

6.33.22. Ticket::Frontend::AgentTicketClose###Priority

Description:	Shows the ticket priority options in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Priority'} =  '0';`

6.33.23. Ticket::Frontend::AgentTicketClose###PriorityDefault

Description:	Defines the default ticket priority in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'PriorityDefault'} =`
	`'3 normal';`

6.33.24. Ticket::Frontend::AgentTicketClose###Title

Description:	Shows the title fields in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose

Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Title'} =  '0';`

6.33.25. Ticket::Frontend::AgentTicketClose###HistoryType

Description:	Defines the history type for the close ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'HistoryType'} = 'AddNote';`

6.33.26. Ticket::Frontend::AgentTicketClose###HistoryComment

Description:	Defines the history comment for the close ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'HistoryComment'} = '%%Close';`

6.33.27. Ticket::Frontend::AgentTicketClose###DynamicField

Description:	Dynamic fields shown in the ticket close screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'DynamicField'} = {};`

6.33.28. Ticket::Frontend::AgentTicketClose###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'RichTextWidth'} = '620';`

6.33.29. Ticket::Frontend::AgentTicketClose###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'RichTextHeight'} = '100';`

6.34. Frontend::Agent::Ticket::ViewCompose

6.34.1. Ticket::Frontend::AgentTicketCompose###Permission

Description:	Required permissions to use the ticket compose screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'Permission'} = 'compose';`

6.34.2. Ticket::Frontend::AgentTicketCompose###RequiredLock

Description:	Defines if a ticket lock is required in the ticket compose screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'RequiredLock'} = '1';`

6.34.3. Ticket::Frontend::AgentTicketCompose###StateDefault

Description:	Defines the default next state of a ticket if it is composed / answered in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'StateDefault'} = 'open';`

6.34.4. Ticket::Frontend::AgentTicketCompose###StateType

Description:	Defines the next possible states after composing / answering a ticket in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'StateType'} =  [`
	`  'open',`
	`  'closed',`
	`  'pending auto',`
	`  'pending reminder'`
	`];`

6.34.5. Ticket::Frontend::AgentTicketCompose###ArticleTypes

Description:	Specifies the different article types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'ArticleTypes'} = [`
	`  'email-external',`
	`  'email-internal'`
	`];`

6.34.6. Ticket::Frontend::ResponseFormat

Description:	Defines the format of responses in the ticket compose screen of the agent interface ([% Data.OrigFrom | html %] is From 1:1, [% Data.OrigFromName | html %] is only realname of From).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::ResponseFormat'} =  '[% Data.Salutation | html %]`
	`[% Data.StdResponse | html %]`
	`[% Data.Signature | html %]`
	
	`[% Data.Created | Localize("TimeShort") %] - [% Data.OrigFromName | html %] [% Translate("wrote") | html %]:`
	`[% Data.Body | html %]`
	`';`

6.34.7. Ticket::Frontend::Quote

Description:	Defines the used character for plaintext email quotes in the ticket compose screen of the agent interface. If this is empty or inactive, original emails will not be quoted but appended to the response.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::Quote'} =  '>';`

6.34.8. Ticket::Frontend::ComposeAddCustomerAddress

Description:	Adds customers email addresses to recipients in the ticket compose screen of the agent interface. The customers email address won't be added if the article type is email-internal.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ComposeAddCustomerAddress'} =  '1';`

6.34.9. Ticket::Frontend::ComposeReplaceSenderAddress

Description:	Replaces the original sender with current customer's email address on compose answer in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ComposeReplaceSenderAddress'} =  '0';`

6.34.10. Ticket::Frontend::ComposeExcludeCcRecipients

Description:	Uses Cc recipients in reply Cc list on compose an email answer in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ComposeExcludeCcRecipients'} =  '0';`

6.34.11. Ticket::Frontend::AgentTicketCompose###DynamicField

Description:	Dynamic fields shown in the ticket compose screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'DynamicField'} = {};`

6.35. Frontend::Agent::Ticket::ViewCustomer

6.35.1. Ticket::Frontend::AgentTicketCustomer###Permission

Description:	Required permissions to change the customer of a ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCustomer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCustomer'}->{'Permission'} = 'customer';`

6.35.2. Ticket::Frontend::AgentTicketCustomer###RequiredLock

Description:	Defines if a ticket lock is required to change the customer of a ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCustomer
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCustomer'}->{'RequiredLock'} = '0';`

6.36. Frontend::Agent::Ticket::ViewEmailNew

6.36.1. Ticket::Frontend::AgentTicketEmail###Priority

Description:	Sets the default priority for new email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Priority'} =  '3 normal';`

6.36.2. Ticket::Frontend::AgentTicketEmail###ArticleType

Description:	Sets the default article type for new email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'ArticleType'} = 'email-external';`

6.36.3. Ticket::Frontend::AgentTicketEmail###SenderType

Description:	Sets the default sender type for new email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'SenderType'} = 'agent';`

6.36.4. Ticket::Frontend::AgentTicketEmail###Subject

Description:	Sets the default subject for new email tickets (e.g. 'email Outbound') in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Subject'} =  '';`

6.36.5. Ticket::Frontend::AgentTicketEmail###Body

Description:	Sets the default text for new email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Body'} =  '';`

6.36.6. Ticket::Frontend::AgentTicketEmail###StateDefault

Description:	Sets the default next ticket state, after the creation of an email ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'StateDefault'} = 'open';`

6.36.7. Ticket::Frontend::AgentTicketEmail###StateType

Description:	Determines the next possible ticket states, after the creation of a new email ticket in the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'StateType'} =  [`
	`  'open',`
	`  'pending auto',`
	`  'pending reminder',`
	`  'closed'`
	`];`

6.36.8. Ticket::Frontend::AgentTicketEmail###HistoryType

Description:	Defines the history type for the email ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'HistoryType'} =`
	`'EmailAgent';`

6.36.9. Ticket::Frontend::AgentTicketEmail###HistoryComment

Description:	Defines the history comment for the email ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'HistoryComment'} =`
	`'';`

6.36.10. Ticket::Frontend::AgentTicketEmail###ServiceMandatory

Description:	Sets if service must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'ServiceMandatory'} =`
	`'0';`

6.36.11. Ticket::Frontend::AgentTicketEmail###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew

Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'SLAMandatory'} = '0';`

6.36.12. Ticket::Frontend::AgentTicketEmail###DynamicField

Description:	Dynamic fields shown in the ticket email screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'DynamicField'} = {};`

6.36.13. Ticket::Frontend::AgentTicketEmail###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'RichTextWidth'} = '620';`

6.36.14. Ticket::Frontend::AgentTicketEmail###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'RichTextHeight'} = '320';`

6.37. Frontend::Agent::Ticket::ViewEscalation

6.37.1. Ticket::Frontend::AgentTicketEscalationView###TicketPermission

Description:	Defines the required permission to show a ticket in the escalation view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'TicketPermission'} =  'rw';`

6.37.2. Ticket::Frontend::AgentTicketEscalationView###ViewableTicketsPage

Description:	Shows all open tickets (even if they are locked) in the escalation view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'ViewableTicketsPage'} =  '50';`

6.37.3. Ticket::Frontend::AgentTicketEscalationView###SortBy::Default

Description:	Defines the default ticket attribute for ticket sorting in the escalation view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'SortBy::Default'} =  'EscalationTime';`

6.37.4. Ticket::Frontend::AgentTicketEscalationView###Order::Default

Description:	Defines the default ticket order (after priority sort) in the escalation view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'Order::Default'} =  'Up';`

6.37.5. Ticket::Frontend::AgentTicketEscalationView###DefaultColumns

Description:	Columns that can be filtered in the escalation view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	0

Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'DefaultColumns'} = {
	  'Age' => '2',
	  'Changed' => '1',
	  'CustomerID' => '2',
	  'CustomerName' => '1',
	  'CustomerUserID' => '1',
	  'EscalationResponseTime' => '1',
	  'EscalationSolutionTime' => '1',
	  'EscalationTime' => '2',
	  'EscalationUpdateTime' => '1',
	  'Lock' => '2',
	  'Owner' => '2',
	  'PendingTime' => '1',
	  'Priority' => '1',
	  'Queue' => '2',
	  'Responsible' => '1',
	  'SLA' => '1',
	  'Service' => '1',
	  'State' => '2',
	  'TicketNumber' => '2',
	  'Title' => '2',
	  'Type' => '1'
	};

6.38. Frontend::Agent::Ticket::ViewForward

6.38.1. Ticket::Frontend::AgentTicketForward###Permission

Description:	Required permissions to use the ticket forward screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketForward'}->{'Permission'} = 'forward';

6.38.2. Ticket::Frontend::AgentTicketForward###RequiredLock

Description:	Defines if a ticket lock is required in the ticket forward screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketForward'}->{'RequiredLock'} = '1';

6.38.3. Ticket::Frontend::AgentTicketForward###StateDefault

Description:	Defines the default next state of a ticket after being forwarded, in the ticket forward screen of the agent interface.
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'StateDefault'} =`
	`'closed successful';`

6.38.4. Ticket::Frontend::AgentTicketForward###StateType

Description:	Defines the next possible states after forwarding a ticket in the ticket forward screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'StateType'} =  [`
	`  'open',`
	`  'closed',`
	`  'pending reminder',`
	`  'pending auto'`
	`];`

6.38.5. Ticket::Frontend::AgentTicketForward###ArticleTypeDefault

Description:	Defines the default type of forwarded message in the ticket forward screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'ArticleTypeDefault'} =  'email-external';`

6.38.6. Ticket::Frontend::AgentTicketForward###ArticleTypes

Description:	Specifies the different article types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'ArticleTypes'} =`
	`[`
	`  'email-external',`
	`  'email-internal'`
	`];`

6.38.7. Ticket::Frontend::AgentTicketForward###DynamicField

Description:	Dynamic fields shown in the ticket forward screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'DynamicField'} = {};`

6.38.8. Ticket::Frontend::AgentTicketForward###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'RichTextWidth'} = '620';`

6.38.9. Ticket::Frontend::AgentTicketForward###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'RichTextHeight'} = '100';`

6.39. Frontend::Agent::Ticket::ViewFreeText

6.39.1. Ticket::Frontend::AgentTicketFreeText###Permission

Description:	Required permissions to use the ticket free text screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Permission'} = 'rw';`

6.39.2. Ticket::Frontend::AgentTicketFreeText###RequiredLock

Description:	Defines if a ticket lock is required in the ticket free text screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'RequiredLock'} = '0';`

6.39.3. Ticket::Frontend::AgentTicketFreeText###TicketType

Description:	Sets the ticket type in the ticket free text screen of the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'TicketType'} = '1';`

6.39.4. Ticket::Frontend::AgentTicketFreeText###Service

Description:	Sets the service in the ticket free text screen of the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Service'} =  '1';`

6.39.5. Ticket::Frontend::AgentTicketFreeText###ServiceMandatory

Description:	Sets if service must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ServiceMandatory'} =  '0';`

6.39.6. Ticket::Frontend::AgentTicketFreeText###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'SLAMandatory'} = '0';`

6.39.7. Ticket::Frontend::AgentTicketFreeText###Queue

Description:	Sets the queue in the ticket free text screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Queue'} =  '0';`

6.39.8. Ticket::Frontend::AgentTicketFreeText###Owner

Description:	Sets the ticket owner in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Owner'} =  '0';`

6.39.9. Ticket::Frontend::AgentTicketFreeText###OwnerMandatory

Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'OwnerMandatory'} =  '0';`

6.39.10. Ticket::Frontend::AgentTicketFreeText###Responsible

Description:	Sets the responsible agent of the ticket in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Responsible'} = '0';`

6.39.11. Ticket::Frontend::AgentTicketFreeText###State

Description:	If a note is added by an agent, sets the state of a ticket in the ticket free text screen of the agent interface.
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'State'} =  '0';`

6.39.12. Ticket::Frontend::AgentTicketFreeText###StateType

Description:	Defines the next state of a ticket after adding a note, in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'StateType'} =  [`
	`  'open',`
	`  'closed',`
	`  'pending reminder',`
	`  'pending auto'`
	`];`

6.39.13. Ticket::Frontend::AgentTicketFreeText###StateDefault

Description:	Defines the default next state of a ticket after adding a note, in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'StateDefault'} =`
	`'open';`

6.39.14. Ticket::Frontend::AgentTicketFreeText###Note

Description:	Allows adding notes in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Note'} =  '0';`

6.39.15. Ticket::Frontend::AgentTicketFreeText###NoteMandatory

Description:	Sets if note must be filled in by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'NoteMandatory'} = '0';`

6.39.16. Ticket::Frontend::AgentTicketFreeText###Subject

Description:	Defines the default subject of a note in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Subject'} =  '[% Translate("Note") | html %]';`

6.39.17. Ticket::Frontend::AgentTicketFreeText###Body

Description:	Defines the default body of a note in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Body'} =  '';`

6.39.18. Ticket::Frontend::AgentTicketFreeText###InvolvedAgent

Description:	Shows a list of all the involved agents on this ticket, in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'InvolvedAgent'} = '0';`

6.39.19. Ticket::Frontend::AgentTicketFreeText###InformAgent

Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0

Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'InformAgent'} = '0';

6.39.20. Ticket::Frontend::AgentTicketFreeText###ArticleTypeDefault

Description:	Defines the default type of the note in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ArticleTypeDefault'} = 'note-internal';

6.39.21. Ticket::Frontend::AgentTicketFreeText###ArticleTypes

Description:	Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0

Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ArticleTypes'} = {
	  'note-external' => '1',
	  'note-internal' => '1',
	  'note-report' => '0'
	};

6.39.22. Ticket::Frontend::AgentTicketFreeText###Priority

Description:	Shows the ticket priority options in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Priority'} = '0';

6.39.23. Ticket::Frontend::AgentTicketFreeText###PriorityDefault

Description:	Defines the default ticket priority in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'PriorityDefault'} =  '3 normal';`

6.39.24. Ticket::Frontend::AgentTicketFreeText###Title

Description:	Shows the title fields in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Title'} =  '1';`

6.39.25. Ticket::Frontend::AgentTicketFreeText###HistoryType

Description:	Defines the history type for the ticket free text screen action, which gets used for ticket history.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'HistoryType'} = 'AddNote';`

6.39.26. Ticket::Frontend::AgentTicketFreeText###HistoryComment

Description:	Defines the history comment for the ticket free text screen action, which gets used for ticket history.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'HistoryComment'} =  '%%FreeText';`

6.39.27. Ticket::Frontend::AgentTicketFreeText###DynamicField

Description:	Dynamic fields shown in the ticket free text screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'DynamicField'} = {};`

6.39.28. Ticket::Frontend::AgentTicketFreeText###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'RichTextWidth'} =`
	` '620';`

6.39.29. Ticket::Frontend::AgentTicketFreeText###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'RichTextHeight'}`
	`=  '100';`

6.40. Frontend::Agent::Ticket::ViewHistory

6.40.1. Ticket::Frontend::HistoryOrder

Description:	Shows the ticket history (reverse ordered) in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewHistory
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::HistoryOrder'} =  'normal';`

6.40.2. Ticket::Frontend::HistoryTypes###000-Framework

Description:	Controls how to display the ticket history entries as readable values.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewHistory
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::HistoryTypes'}->{'000-Framework'} =  {`
	` 'AddNote' => 'Added note (%s)',`
	` 'ArchiveFlagUpdate' => 'Archive state changed: "%s"',`
	` 'Bounce' => 'Bounced to "%s".',`
	` 'CustomerUpdate' => 'Updated: %s',`
	` 'EmailAgent' => 'Email sent to customer.',`
	` 'EmailCustomer' => 'Added email. %s',`
	` 'EscalationResponseTimeNotifyBefore' => 'Escalation response time`
	`forewarned',`

```
  'EscalationResponseTimeStart' => 'Escalation response time in
 effect',
  'EscalationResponseTimeStop' => 'Escalation response time finished',
  'EscalationSolutionTimeNotifyBefore' => 'Escalation solution time
 forewarned',
  'EscalationSolutionTimeStart' => 'Escalation solution time in
 effect',
  'EscalationSolutionTimeStop' => 'Escalation solution time finished',
  'EscalationUpdateTimeNotifyBefore' => 'Escalation update time
 forewarned',
  'EscalationUpdateTimeStart' => 'Escalation update time in effect',
  'EscalationUpdateTimeStop' => 'Escalation update time finished',
  'FollowUp' => 'FollowUp for [%s]. %s',
  'Forward' => 'Forwarded to "%s".',
  'Lock' => 'Locked ticket.',
  'LoopProtection' => 'Loop-Protection! No auto-response sent to
 "%s".',
  'Misc' => '%s',
  'Move' => 'Ticket moved into Queue "%s" (%s) from Queue "%s" (%s).',
  'NewTicket' => 'New Ticket [%s] created (Q=%s;P=%s;S=%s).',
  'OwnerUpdate' => 'New owner is "%s" (ID=%s).',
  'PhoneCallAgent' => 'Agent called customer.',
  'PhoneCallCustomer' => 'Customer called us.',
  'PriorityUpdate' => 'Changed priority from "%s" (%s) to "%s" (%s).',
  'Remove' => '%s',
  'ResponsibleUpdate' => 'New responsible is "%s" (ID=%s).',
  'SLAUpdate' => 'Updated SLA to %s (ID=%s).',
  'SendAgentNotification' => '"%s"-notification sent to "%s".',
  'SendAnswer' => 'Email sent to "%s".',
  'SendAutoFollowUp' => 'AutoFollowUp sent to "%s".',
  'SendAutoReject' => 'AutoReject sent to "%s".',
  'SendAutoReply' => 'AutoReply sent to "%s".',
  'SendCustomerNotification' => 'Notification sent to "%s".',
  'ServiceUpdate' => 'Updated Service to %s (ID=%s).',
  'SetPendingTime' => 'Updated: %s',
  'StateUpdate' => 'Old: "%s" New: "%s"',
  'Subscribe' => 'Added subscription for user "%s".',
  'SystemRequest' => 'System Request (%s).',
  'TicketDynamicFieldUpdate' => 'Updated: %s=%s;%s=%s;%s=%s;',
  'TicketLinkAdd' => 'Added link to ticket "%s".',
  'TicketLinkDelete' => 'Deleted link to ticket "%s".',
  'TimeAccounting' => '%s time unit(s) accounted. Now total %s time
 unit(s).',
  'TitleUpdate' => 'Title updated: Old: "%s", New: "%s"',
  'TypeUpdate' => 'Updated Type to %s (ID=%s).',
  'Unlock' => 'Unlocked ticket.',
  'Unsubscribe' => 'Removed subscription for user "%s".',
  'WebRequestCustomer' => 'Customer request via web.'
};
```

6.41. Frontend::Agent::Ticket::ViewLocked

6.41.1. Ticket::Frontend::AgentTicketLockedView###SortBy::Default

Description:	Defines the default ticket attribute for ticket sorting in the locked ticket view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewLocked
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketLockedView'}->{'SortBy::Default'} =  'Age';`

6.41.2. Ticket::Frontend::AgentTicketLockedView###Order::Default

Description:	Defines the default ticket order in the ticket locked view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewLocked
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketLockedView'}->{'Order::Default'} =  'Up';`

6.41.3. Ticket::Frontend::AgentTicketLockedView###DefaultColumns

Description:	Columns that can be filtered in the locked view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewLocked
Valid:	1
Required:	0
Config-Setting:	see below

```
 $Self->{'Ticket::Frontend::AgentTicketLockedView'}-
>{'DefaultColumns'} =  {
  'Age' => '2',
  'Changed' => '1',
  'CustomerID' => '2',
  'CustomerName' => '1',
  'CustomerUserID' => '1',
  'EscalationResponseTime' => '1',
  'EscalationSolutionTime' => '1',
  'EscalationTime' => '1',
  'EscalationUpdateTime' => '1',
  'Lock' => '2',
  'Owner' => '2',
  'PendingTime' => '1',
  'Priority' => '1',
  'Queue' => '2',
  'Responsible' => '1',
  'SLA' => '1',
  'Service' => '1',
  'State' => '2',
  'TicketNumber' => '2',
  'Title' => '2',
  'Type' => '1'
};
```

6.42. Frontend::Agent::Ticket::ViewMerge

6.42.1. Ticket::Frontend::AgentTicketMerge###Permission

Description:	Required permissions to use the ticket merge screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'Permission'} = 'rw';`

6.42.2. Ticket::Frontend::AgentTicketMerge###RequiredLock

Description:	Defines if a ticket lock is required in the ticket merge screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'RequiredLock'} = '1';`

6.42.3. Ticket::Frontend::MergeText

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:	`$Self->{'Ticket::Frontend::MergeText'} =  'Your email with ticket number "<OTRS_TICKET>" is merged to "<OTRS_MERGE_TO_TICKET>".';`

6.42.4. Ticket::Frontend::AutomaticMergeSubject

Description:	When tickets are merged, a note will be added automatically to the ticket which is no longer active. Here you can define the subject of this note (this subject cannot be changed by the agent).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AutomaticMergeSubject'} =  'Ticket Merged';`

6.42.5. Ticket::Frontend::AutomaticMergeText

Description:	When tickets are merged, a note will be added automatically to the ticket which is no longer active. Here you can define the body of this note (this text cannot be changed by the agent).
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AutomaticMergeText'} =  'Merged Ticket <OTRS_TICKET> to <OTRS_MERGE_TO_TICKET>.';`

6.42.6. Ticket::Frontend::AgentTicketMerge###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'RichTextWidth'} = '620';`

6.42.7. Ticket::Frontend::AgentTicketMerge###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'RichTextHeight'} = '100';`

6.43. Frontend::Agent::Ticket::ViewMove

6.43.1. Ticket::Frontend::MoveType

Description:	Determines if the list of possible queues to move to ticket into should be displayed in a dropdown list or in a new window in the agent interface. If "New Window" is set you can add a move note to the ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::MoveType'} =  'form';`

6.43.2. Ticket::Frontend::AgentTicketMove###RequiredLock

Description:	Automatically lock and set owner to current Agent after opening the move ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove

Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'RequiredLock'} =`
	`'1';`

6.43.3. Ticket::Frontend::AgentTicketMove###State

Description:	Allows to set a new ticket state in the move ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'State'} =  '1';`

6.43.4. Ticket::Frontend::AgentTicketMove###StateType

Description:	Defines the next state of a ticket after being moved to another queue, in the move ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'StateType'} =  [`
	`  'open',`
	`  'closed'`
	`];`

6.43.5. Ticket::Frontend::AgentTicketMove###Priority

Description:	Shows the ticket priority options in the move ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Priority'} =  '0';`

6.43.6. Ticket::Frontend::AgentTicketMove###Note

Description:	Allows adding notes in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Note'} =  '0';`

6.43.7. Ticket::Frontend::AgentTicketMove###NoteMandatory

Description:	Sets if note must be filled in by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'NoteMandatory'} = '0';`

6.43.8. Ticket::Frontend::AgentTicketMove###NextScreen

Description:	Determines the next screen after the ticket is moved. LastScreenOverview will return the last overview screen (e.g. search results, queueview, dashboard). TicketZoom will return to the TicketZoom.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'NextScreen'} = 'TicketZoom';`

6.43.9. Ticket::Frontend::AgentTicketMove###Subject

Description:	Sets the default subject for notes added in the ticket move screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Subject'} =  '[% Translate("Change Queue") | html %]';`

6.43.10. Ticket::Frontend::AgentTicketMove###Body

Description:	Sets the default body text for notes added in the ticket move screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Body'} =  '';`

6.43.11. Ticket::Frontend::AgentTicketMove###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'RichTextWidth'} = '620';`

6.43.12. Ticket::Frontend::AgentTicketMove###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'RichTextHeight'} = '100';`

6.43.13. Ticket::Frontend::AgentTicketMove###DynamicField

Description:	Dynamic fields shown in the ticket move screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'DynamicField'} =  {};`

6.44. Frontend::Agent::Ticket::ViewNote

6.44.1. Ticket::Frontend::AgentTicketNote###Permission

Description:	Required permissions to use the ticket note screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Permission'} = 'note';`

6.44.2. Ticket::Frontend::AgentTicketNote###RequiredLock

Description:	Defines if a ticket lock is required in the ticket note screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'RequiredLock'} = '0';`

6.44.3. Ticket::Frontend::AgentTicketNote###TicketType

Description:	Sets the ticket type in the ticket note screen of the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'TicketType'} =  '0';`

6.44.4. Ticket::Frontend::AgentTicketNote###Service

Description:	Sets the service in the ticket note screen of the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Service'} =  '0';`

6.44.5. Ticket::Frontend::AgentTicketNote###ServiceMandatory

Description:	Sets if service must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'ServiceMandatory'} = '0';`

6.44.6. Ticket::Frontend::AgentTicketNote###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'SLAMandatory'} = '0';`

6.44.7. Ticket::Frontend::AgentTicketNote###Queue

Description:	Sets the queue in the ticket note screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Queue'} =  '0';`

6.44.8. Ticket::Frontend::AgentTicketNote###Owner

Description:	Sets the ticket owner in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Owner'} =  '0';`

6.44.9. Ticket::Frontend::AgentTicketNote###OwnerMandatory

Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'OwnerMandatory'} = '0';`

6.44.10. Ticket::Frontend::AgentTicketNote###Responsible

Description:	Sets the responsible agent of the ticket in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Responsible'} =  '0';`

6.44.11. Ticket::Frontend::AgentTicketNote###State

Description:	If a note is added by an agent, sets the state of a ticket in the ticket note screen of the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'State'} =  '0';`

6.44.12. Ticket::Frontend::AgentTicketNote###StateType

Description:	Defines the next state of a ticket after adding a note, in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'StateType'} =  [`
	`  'open',`
	`  'closed',`
	`  'pending reminder',`
	`  'pending auto'`
	`];`

6.44.13. Ticket::Frontend::AgentTicketNote###StateDefault

Description:	Defines the default next state of a ticket after adding a note, in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'StateDefault'} =  'open';`

6.44.14. Ticket::Frontend::AgentTicketNote###Note

Description:	Allows adding notes in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Note'} =  '1';`

6.44.15. Ticket::Frontend::AgentTicketNote###NoteMandatory

Description:	Sets if note must be filled in by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'NoteMandatory'} = '1';`

6.44.16. Ticket::Frontend::AgentTicketNote###Subject

Description:	Sets the default subject for notes added in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Subject'} =  '[% Translate("Note") | html %]';`

6.44.17. Ticket::Frontend::AgentTicketNote###Body

Description:	Sets the default body text for notes added in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Body'} =  '';`

6.44.18. Ticket::Frontend::AgentTicketNote###InvolvedAgent

Description:	Shows a list of all the involved agents on this ticket, in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'InvolvedAgent'} = '0';`

6.44.19. Ticket::Frontend::AgentTicketNote###InformAgent

Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'InformAgent'} =  '0';`

6.44.20. Ticket::Frontend::AgentTicketNote###ArticleTypeDefault

Description:	Defines the default type of the note in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'ArticleTypeDefault'} =  'note-internal';`

6.44.21. Ticket::Frontend::AgentTicketNote###ArticleTypes

Description:	Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'ArticleTypes'} =  {`
	`  'note-external' => '1',`
	`  'note-internal' => '1',`
	`  'note-report' => '0'`
	`};`

6.44.22. Ticket::Frontend::AgentTicketNote###Priority

Description:	Shows the ticket priority options in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Priority'} =  '0';`

6.44.23. Ticket::Frontend::AgentTicketNote###PriorityDefault

Description:	Defines the default ticket priority in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'PriorityDefault'} =  '3 normal';`

6.44.24. Ticket::Frontend::AgentTicketNote###Title

Description:	Shows the title fields in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Title'} =  '0';`

6.44.25. Ticket::Frontend::AgentTicketNote###HistoryType

Description:	Defines the history type for the ticket note screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'HistoryType'} = 'AddNote';`

6.44.26. Ticket::Frontend::AgentTicketNote###HistoryComment

Description:	Defines the history comment for the ticket note screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'HistoryComment'} = '%%Note';`

6.44.27. Ticket::Frontend::AgentTicketNote###DynamicField

Description:	Dynamic fields shown in the ticket note screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'DynamicField'} =  {};`

6.44.28. Ticket::Frontend::AgentTicketNote###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'RichTextWidth'} = '620';`

6.44.29. Ticket::Frontend::AgentTicketNote###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'RichTextHeight'} = '100';`

6.45. Frontend::Agent::Ticket::ViewOwner

6.45.1. Ticket::Frontend::AgentTicketOwner###Permission

Description:	Required permissions to use the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Permission'} = 'owner';`

6.45.2. Ticket::Frontend::AgentTicketOwner###RequiredLock

Description:	Defines if a ticket lock is required in the ticket owner screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'RequiredLock'} = '0';`

6.45.3. Ticket::Frontend::AgentTicketOwner###TicketType

Description:	Sets the ticket type in the ticket owner screen of a zoomed ticket in the agent interface (Ticket::Type needs to be activated).

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'TicketType'} =  '0';`

6.45.4. Ticket::Frontend::AgentTicketOwner###Service

Description:	Sets the service in the ticket owner screen of a zoomed ticket in the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Service'} =  '0';`

6.45.5. Ticket::Frontend::AgentTicketOwner###ServiceMandatory

Description:	Sets if service must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'ServiceMandatory'} = '0';`

6.45.6. Ticket::Frontend::AgentTicketOwner###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'SLAMandatory'} = '0';`

6.45.7. Ticket::Frontend::AgentTicketOwner###Queue

Description:	Sets the queue in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Queue'} =  '0';`

6.45.8. Ticket::Frontend::AgentTicketOwner###Owner

Description:	Sets the ticket owner in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Owner'} =  '1';`

6.45.9. Ticket::Frontend::AgentTicketOwner###OwnerMandatory

Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'OwnerMandatory'} = '1';`

6.45.10. Ticket::Frontend::AgentTicketOwner###Responsible

Description:	Sets the responsible agent of the ticket in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Responsible'} = '0';`

6.45.11. Ticket::Frontend::AgentTicketOwner###State

Description:	If a note is added by an agent, sets the state of the ticket in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'State'} =  '0';`

6.45.12. Ticket::Frontend::AgentTicketOwner###StateType

Description:	Defines the next state of a ticket after adding a note, in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'StateType'} =  [`
	`  'open',`
	`  'pending reminder',`
	`  'pending auto'`
	`];`

6.45.13. Ticket::Frontend::AgentTicketOwner###StateDefault

Description:	Defines the default next state of a ticket after adding a note, in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'StateDefault'} =`
	`'open';`

6.45.14. Ticket::Frontend::AgentTicketOwner###Note

Description:	Allows adding notes in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Note'} =  '1';`

6.45.15. Ticket::Frontend::AgentTicketOwner###NoteMandatory

Description:	Sets if note must be filled in by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'NoteMandatory'} =`
	`'1';`

6.45.16. Ticket::Frontend::AgentTicketOwner###Subject

Description:	Sets the default subject for notes added in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Subject'} = '[% Translate("Owner Update") | html %]!';`

6.45.17. Ticket::Frontend::AgentTicketOwner###Body

Description:	Sets the default body text for notes added in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Body'} = '';`

6.45.18. Ticket::Frontend::AgentTicketOwner###InvolvedAgent

Description:	Shows a list of all the involved agents on this ticket, in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'InvolvedAgent'} = '0';`

6.45.19. Ticket::Frontend::AgentTicketOwner###InformAgent

Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'InformAgent'} = '0';`

6.45.20. Ticket::Frontend::AgentTicketOwner###ArticleTypeDefault

Description:	Defines the default type of the note in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'ArticleTypeDefault'}`
	`=  'note-internal';`

6.45.21. Ticket::Frontend::AgentTicketOwner###ArticleTypes

Description:	Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'ArticleTypes'} =  {`
	`  'note-external' => '0',`
	`  'note-internal' => '1',`
	`  'note-report' => '0'`
	`};`

6.45.22. Ticket::Frontend::AgentTicketOwner###Priority

Description:	Shows the ticket priority options in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Priority'} =  '0';`

6.45.23. Ticket::Frontend::AgentTicketOwner###PriorityDefault

Description:	Defines the default ticket priority in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'PriorityDefault'} =`
	`'3 normal';`

6.45.24. Ticket::Frontend::AgentTicketOwner###Title

Description:	Shows the title fields in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Title'} =  '0';`

6.45.25. Ticket::Frontend::AgentTicketOwner###HistoryType

Description:	Defines the history type for the ticket owner screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'HistoryType'} = 'AddNote';`

6.45.26. Ticket::Frontend::AgentTicketOwner###HistoryComment

Description:	Defines the history comment for the ticket owner screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'HistoryComment'} = '%%Owner';`

6.45.27. Ticket::Frontend::AgentTicketOwner###DynamicField

Description:	Dynamic fields shown in the ticket owner screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'DynamicField'} = {};`

6.45.28. Ticket::Frontend::AgentTicketOwner###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'RichTextWidth'} = '620';`

6.45.29. Ticket::Frontend::AgentTicketOwner###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'RichTextHeight'} = '100';`

6.46. Frontend::Agent::Ticket::ViewPending

6.46.1. Ticket::Frontend::AgentTicketPending###Permission

Description:	Required permissions to use the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Permission'} = 'pending';`

6.46.2. Ticket::Frontend::AgentTicketPending###RequiredLock

Description:	Defines if a ticket lock is required in the ticket pending screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'RequiredLock'} = '1';`

6.46.3. Ticket::Frontend::AgentTicketPending###TicketType

Description:	Sets the ticket type in the ticket pending screen of a zoomed ticket in the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'TicketType'} = '0';`

6.46.4. Ticket::Frontend::AgentTicketPending###Service

Description:	Sets the service in the ticket pending screen of a zoomed ticket in the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Service'} =  '0';`

6.46.5. Ticket::Frontend::AgentTicketPending###ServiceMandatory

Description:	Sets if service must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'ServiceMandatory'} =  '0';`

6.46.6. Ticket::Frontend::AgentTicketPending###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'SLAMandatory'} = '0';`

6.46.7. Ticket::Frontend::AgentTicketPending###Queue

Description:	Sets the queue in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Queue'} =  '0';`

6.46.8. Ticket::Frontend::AgentTicketPending###Owner

Description:	Sets the ticket owner in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending

Valid:	1
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Owner'} = '0';

6.46.9. Ticket::Frontend::AgentTicketPending###OwnerMandatory

Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketPending'}->{'OwnerMandatory'} = '0';

6.46.10. Ticket::Frontend::AgentTicketPending###Responsible

Description:	Sets the responsible agent of the ticket in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Responsible'} = '0';

6.46.11. Ticket::Frontend::AgentTicketPending###State

Description:	If a note is added by an agent, sets the state of the ticket in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketPending'}->{'State'} = '1';

6.46.12. Ticket::Frontend::AgentTicketPending###StateType

Description:	Defines the next state of a ticket after adding a note, in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	1

Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketPending'}->{'StateType'} = [
	  'pending reminder',
	  'pending auto'
	];

6.46.13. Ticket::Frontend::AgentTicketPending###StateDefault

Description:	Defines the default next state of a ticket after adding a note, in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'StateDefault'} = 'pending reminder';`

6.46.14. Ticket::Frontend::AgentTicketPending###Note

Description:	Allows adding notes in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Note'} =  '1';`

6.46.15. Ticket::Frontend::AgentTicketPending###NoteMandatory

Description:	Sets if note must be filled in by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'NoteMandatory'} = '1';`

6.46.16. Ticket::Frontend::AgentTicketPending###Subject

Description:	Sets the default subject for notes added in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Subject'} =  '[% Translate("Pending") | html %]!';`

6.46.17. Ticket::Frontend::AgentTicketPending###Body

Description:	Sets the default body text for notes added in the ticket pending screen of a zoomed ticket in the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Body'} =  '';`

6.46.18. Ticket::Frontend::AgentTicketPending###InvolvedAgent

Description:	Shows a list of all the involved agents on this ticket, in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'InvolvedAgent'} = '0';`

6.46.19. Ticket::Frontend::AgentTicketPending###InformAgent

Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'InformAgent'} = '0';`

6.46.20. Ticket::Frontend::AgentTicketPending###ArticleTypeDefault

Description:	Defines the default type of the note in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'ArticleTypeDefault'} =  'note-internal';`

6.46.21. Ticket::Frontend::AgentTicketPending###ArticleTypes

Description:	Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'ArticleTypes'} =`
	`{`
	`  'note-external' => '0',`
	`  'note-internal' => '1',`
	`  'note-report' => '0'`
	`};`

6.46.22. Ticket::Frontend::AgentTicketPending###Priority

Description:	Shows the ticket priority options in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Priority'} =  '0';`

6.46.23. Ticket::Frontend::AgentTicketPending###PriorityDefault

Description:	Defines the default ticket priority in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'PriorityDefault'}`
	`=  '3 normal';`

6.46.24. Ticket::Frontend::AgentTicketPending###Title

Description:	Shows the title fields in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Title'} =  '0';`

6.46.25. Ticket::Frontend::AgentTicketPending###HistoryType

Description:	Defines the history type for the ticket pending screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending

Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'HistoryType'} = 'AddNote';`

6.46.26. Ticket::Frontend::AgentTicketPending###HistoryComment

Description:	Defines the history comment for the ticket pending screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'HistoryComment'} = '%%Pending';`

6.46.27. Ticket::Frontend::AgentTicketPending###DynamicField

Description:	Dynamic fields shown in the ticket pending screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'DynamicField'} = {};`

6.46.28. Ticket::Frontend::AgentTicketPending###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'RichTextWidth'} = '620';`

6.46.29. Ticket::Frontend::AgentTicketPending###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'RichTextHeight'} = '100';`

6.47. Frontend::Agent::Ticket::ViewPhoneInbound

6.47.1. Ticket::Frontend::AgentTicketPhoneInbound###Permission

Description:	Required permissions to use the ticket phone inbound screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'Permission'} = 'phone';`

6.47.2. Ticket::Frontend::AgentTicketPhoneInbound###RequiredLock

Description:	Defines if a ticket lock is required in the ticket phone inbound screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'RequiredLock'} = '0';`

6.47.3. Ticket::Frontend::AgentTicketPhoneInbound###ArticleType

Description:	Defines the default type of the note in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'ArticleType'} = 'phone';`

6.47.4. Ticket::Frontend::AgentTicketPhoneInbound###SenderType

Description:	Defines the default sender type for phone tickets in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'SenderType'} =  'customer';`

6.47.5. Ticket::Frontend::AgentTicketPhoneInbound###Subject

Description:	Defines the default subject for phone tickets in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'Subject'} = '[% Translate("Phone call") | html %]!';`

6.47.6. Ticket::Frontend::AgentTicketPhoneInbound###Body

Description:	Defines the default note body text for phone tickets in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'Body'} =  '';`

6.47.7. Ticket::Frontend::AgentTicketPhoneInbound###State

Description:	Defines the default ticket next state after adding a phone note in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'State'} = 'open';`

6.47.8. Ticket::Frontend::AgentTicketPhoneInbound###StateType

Description:	Next possible ticket states after adding a phone note in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'StateType'} =`
	`  [`

	`'open',`
	`'pending auto',`
	`'pending reminder',`
	`'closed'`
	`];`

6.47.9. Ticket::Frontend::AgentTicketPhoneInbound###HistoryType

Description:	Defines the history type for the ticket phone inbound screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'HistoryType'} =  'PhoneCallCustomer';`

6.47.10. Ticket::Frontend::AgentTicketPhoneInbound###HistoryComment

Description:	Defines the history comment for the ticket phone inbound screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'HistoryComment'} =  '';`

6.47.11. Ticket::Frontend::AgentTicketPhoneInbound###DynamicField

Description:	Dynamic fields shown in the ticket phone inbound screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'DynamicField'} =  {};`

6.47.12. Ticket::Frontend::AgentTicketPhoneInbound###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'RichTextWidth'} =  '475';`

6.47.13. Ticket::Frontend::AgentTicketPhoneInbound###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'RichTextHeight'} =  '200';`

6.48. Frontend::Agent::Ticket::ViewPhoneNew

6.48.1. Ticket::Frontend::AgentTicketPhone###Priority

Description:	Sets the default priority for new phone tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'Priority'} =  '3 normal';`

6.48.2. Ticket::Frontend::AgentTicketPhone###ArticleType

Description:	Sets the default article type for new phone tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'ArticleType'} = 'phone';`

6.48.3. Ticket::Frontend::AgentTicketPhone###SenderType

Description:	Sets the default sender type for new phone ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'SenderType'} = 'customer';`

6.48.4. Ticket::Frontend::AgentTicketPhone::AllowMultipleFrom

Description:	Controls if more than one from entry can be set in the new phone ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone::AllowMultipleFrom'} = '1';`

6.48.5. Ticket::Frontend::AgentTicketPhone###Subject

Description:	Sets the default subject for new phone tickets (e.g. 'Phone call') in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'Subject'} =  '';`

6.48.6. Ticket::Frontend::AgentTicketPhone###Body

Description:	Sets the default note text for new telephone tickets. E.g 'New ticket via call' in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'Body'} =  '';`

6.48.7. Ticket::Frontend::AgentTicketPhone###StateDefault

Description:	Sets the default next state for new phone tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'StateDefault'} = 'open';`

6.48.8. Ticket::Frontend::AgentTicketPhone###StateType

Description:	Determines the next possible ticket states, after the creation of a new phone ticket in the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'StateType'} =  [`
	`  'open',`
	`  'pending auto',`
	`  'pending reminder',`
	`  'closed'`
	`];`

6.48.9. Ticket::Frontend::AgentTicketPhone###HistoryType

Description:	Defines the history type for the phone ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'HistoryType'} =`
	`'PhoneCallCustomer';`

6.48.10. Ticket::Frontend::AgentTicketPhone###HistoryComment

Description:	Defines the history comment for the phone ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'HistoryComment'} =`
	`'';`

6.48.11. Ticket::Frontend::AgentTicketPhone###SplitLinkType

Description:	Sets the default link type of splitted tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'SplitLinkType'} =  {`
	`  'Direction' => 'Target',`
	`  'LinkType' => 'ParentChild'`
	`};`

6.48.12. Ticket::Frontend::AgentTicketPhone###ServiceMandatory

Description:	Sets if service must be selected by the agent.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'ServiceMandatory'} = '0';`

6.48.13. Ticket::Frontend::AgentTicketPhone###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'SLAMandatory'} = '0';`

6.48.14. Ticket::Frontend::AgentTicketPhone###DynamicField

Description:	Dynamic fields shown in the ticket phone screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'DynamicField'} = {};`

6.48.15. Ticket::Frontend::AgentTicketPhone###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'RichTextWidth'} = '620';`

6.48.16. Ticket::Frontend::AgentTicketPhone###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew

Valid:	1
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'RichTextHeight'} = '320';

6.49. Frontend::Agent::Ticket::ViewPhoneOutbound

6.49.1. Ticket::Frontend::AgentTicketPhoneOutbound###Permission

Description:	Required permissions to use the ticket phone outbound screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'Permission'} = 'phone';

6.49.2. Ticket::Frontend::AgentTicketPhoneOutbound###RequiredLock

Description:	Defines if a ticket lock is required in the ticket phone outbound screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'RequiredLock'} = '1';

6.49.3. Ticket::Frontend::AgentTicketPhoneOutbound###ArticleType

Description:	Defines the default type of the note in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'ArticleType'} = 'phone';

6.49.4. Ticket::Frontend::AgentTicketPhoneOutbound###SenderType

Description:	Defines the default sender type for phone tickets in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound

Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'SenderType'} = 'agent';`

6.49.5. Ticket::Frontend::AgentTicketPhoneOutbound###Subject

Description:	Defines the default subject for phone tickets in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'Subject'} = '[% Translate("Phone call") | html %]!';`

6.49.6. Ticket::Frontend::AgentTicketPhoneOutbound###Body

Description:	Defines the default note body text for phone tickets in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'Body'} = '';`

6.49.7. Ticket::Frontend::AgentTicketPhoneOutbound###State

Description:	Defines the default ticket next state after adding a phone note in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'State'} = 'closed successful';`

6.49.8. Ticket::Frontend::AgentTicketPhoneOutbound###StateType

Description:	Next possible ticket states after adding a phone note in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'StateType'} =  [`
	`'open',`
	`'pending auto',`
	`'pending reminder',`
	`'closed'`
	`];`

6.49.9. Ticket::Frontend::AgentTicketPhoneOutbound###HistoryType

Description:	Defines the history type for the ticket phone outbound screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'HistoryType'} =  'PhoneCallAgent';`

6.49.10. Ticket::Frontend::AgentTicketPhoneOutbound###HistoryComment

Description:	Defines the history comment for the ticket phone outbound screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'HistoryComment'} =  '';`

6.49.11. Ticket::Frontend::AgentTicketPhoneOutbound###DynamicField

Description:	Dynamic fields shown in the ticket phone outbound screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'DynamicField'} =  {};`

6.49.12. Ticket::Frontend::AgentTicketPhoneOutbound###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'RichTextWidth'} =  '475';`

6.49.13. Ticket::Frontend::AgentTicketPhoneOutbound###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'RichTextHeight'} =  '200';`

6.50. Frontend::Agent::Ticket::ViewPrint

6.50.1. Ticket::Frontend::AgentTicketPrint###DynamicField

Description:	Dynamic fields shown in the ticket print screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPrint
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPrint'}->{'DynamicField'} = {};`

6.51. Frontend::Agent::Ticket::ViewPriority

6.51.1. Ticket::Frontend::AgentTicketPriority###Permission

Description:	Required permissions to use the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Permission'} = 'priority';`

6.51.2. Ticket::Frontend::AgentTicketPriority###RequiredLock

Description:	Defines if a ticket lock is required in the ticket priority screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'RequiredLock'} = '1';`

6.51.3. Ticket::Frontend::AgentTicketPriority###TicketType

Description:	Sets the ticket type in the ticket priority screen of a zoomed ticket in the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'TicketType'} = '0';`

6.51.4. Ticket::Frontend::AgentTicketPriority###Service

Description:	Sets the service in the ticket priority screen of a zoomed ticket in the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Service'} =  '0';`

6.51.5. Ticket::Frontend::AgentTicketPriority###ServiceMandatory

Description:	Sets if service must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ServiceMandatory'} =  '0';`

6.51.6. Ticket::Frontend::AgentTicketPriority###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'SLAMandatory'} = '0';`

6.51.7. Ticket::Frontend::AgentTicketPriority###Queue

Description:	Sets the queue in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Queue'} =  '0';`

6.51.8. Ticket::Frontend::AgentTicketPriority###Owner

Description:	Sets the ticket owner in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Owner'} =  '0';`

6.51.9. Ticket::Frontend::AgentTicketPriority###OwnerMandatory

Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'OwnerMandatory'} =  '0';`

6.51.10. Ticket::Frontend::AgentTicketPriority###Responsible

Description:	Sets the responsible agent of the ticket in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Responsible'} = '0';`

6.51.11. Ticket::Frontend::AgentTicketPriority###State

Description:	If a note is added by an agent, sets the state of the ticket in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'State'} =  '0';`

6.51.12. Ticket::Frontend::AgentTicketPriority###StateType

Description:	Defines the next state of a ticket after adding a note, in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'StateType'} =  [`
	`  'open',`
	`  'pending reminder',`
	`  'pending auto'`
	`];`

6.51.13. Ticket::Frontend::AgentTicketPriority###StateDefault

Description:	Defines the default next state of a ticket after adding a note, in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'StateDefault'} =  'open';`

6.51.14. Ticket::Frontend::AgentTicketPriority###Note

Description:	Allows adding notes in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Note'} =  '1';`

6.51.15. Ticket::Frontend::AgentTicketPriority###NoteMandatory

Description:	Sets if note must be filled in by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'NoteMandatory'} = '1';`

6.51.16. Ticket::Frontend::AgentTicketPriority###Subject

Description:	Sets the default subject for notes added in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Subject'} =  '[% Translate("Priority Update") | html %]!';`

6.51.17. Ticket::Frontend::AgentTicketPriority###Body

Description:	Sets the default body text for notes added in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Body'} =  '';`

6.51.18. Ticket::Frontend::AgentTicketPriority###InvolvedAgent

Description:	Shows a list of all the involved agents on this ticket, in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'InvolvedAgent'} = '0';`

6.51.19. Ticket::Frontend::AgentTicketPriority###InformAgent

Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'InformAgent'} = '0';`

6.51.20. Ticket::Frontend::AgentTicketPriority###ArticleTypeDefault

Description:	Defines the default type of the note in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ArticleTypeDefault'} =  'note-internal';`

6.51.21. Ticket::Frontend::AgentTicketPriority###ArticleTypes

Description:	Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ArticleTypes'} =`
	`{`
	`  'note-external' => '0',`
	`  'note-internal' => '1',`
	`  'note-report' => '0'`
	`};`

6.51.22. Ticket::Frontend::AgentTicketPriority###Priority

Description:	Shows the ticket priority options in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Priority'} = '1';`

6.51.23. Ticket::Frontend::AgentTicketPriority###PriorityDefault

Description:	Defines the default ticket priority in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	0

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'PriorityDefault'} =  '3 normal';`

6.51.24. Ticket::Frontend::AgentTicketPriority###Title

Description:	Shows the title fields in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Title'} =  '0';`

6.51.25. Ticket::Frontend::AgentTicketPriority###HistoryType

Description:	Defines the history type for the ticket priority screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'HistoryType'} = 'AddNote';`

6.51.26. Ticket::Frontend::AgentTicketPriority###HistoryComment

Description:	Defines the history comment for the ticket priority screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'HistoryComment'} =  '%%Priority';`

6.51.27. Ticket::Frontend::AgentTicketPriority###DynamicField

Description:	Dynamic fields shown in the ticket priority screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'DynamicField'} = {};`

6.51.28. Ticket::Frontend::AgentTicketPriority###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'RichTextWidth'} = '620';`

6.51.29. Ticket::Frontend::AgentTicketPriority###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'RichTextHeight'} = '100';`

6.52. Frontend::Agent::Ticket::ViewQueue

6.52.1. Ticket::Frontend::AgentTicketQueue###StripEmptyLines

Description:	Strips empty lines on the ticket preview in the queue view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'StripEmptyLines'} = '0';`

6.52.2. Ticket::Frontend::AgentTicketQueue###ViewAllPossibleTickets

Description:	Shows all both ro and rw queues in the queue view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'ViewAllPossibleTickets'} = '0';`

6.52.3. Ticket::Frontend::AgentTicketQueue###HighlightAge1

Description:	Sets the age in minutes (first level) for highlighting queues that contain untouched tickets.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'HighlightAge1'} = '1440';`

6.52.4. Ticket::Frontend::AgentTicketQueue###HighlightAge2

Description:	Sets the age in minutes (second level) for highlighting queues that contain untouched tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'HighlightAge2'} = '2880';`

6.52.5. Ticket::Frontend::AgentTicketQueue###Blink

Description:	Activates a blinking mechanism of the queue that contains the oldest ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'Blink'} =  '1';`

6.52.6. Ticket::Frontend::AgentTicketQueue###QueueSort

Description:	Sorts the tickets (ascendingly or descendingly) when a single queue is selected in the queue view and after the tickets are sorted by priority. Values: 0 = ascending (oldest on top, default), 1 = descending (youngest on top). Use the QueueID for the key and 0 or 1 for value.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'QueueSort'} =  {`
	`  '3' => '0',`
	`  '7' => '1'`
	`};`

6.52.7. Ticket::Frontend::AgentTicketQueue###SortBy::Default

Description:	Defines the default sort criteria for all queues displayed in the queue view.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'SortBy::Default'} =`
	`'Age';`

6.52.8. Ticket::Frontend::AgentTicketQueue###PreSort::ByPriority

Description:	Defines if a pre-sorting by priority should be done in the queue view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}-`
	`>{'PreSort::ByPriority'} =  '1';`

6.52.9. Ticket::Frontend::AgentTicketQueue###Order::Default

Description:	Defines the default sort order for all queues in the queue view, after priority sort.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'Order::Default'} =`
	`'Up';`

6.52.10. Ticket::Frontend::AgentTicketQueue###DefaultColumns

Description:	Columns that can be filtered in the queue view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'DefaultColumns'} =`
	`{`
	` 'Age' => '2',`
	` 'Changed' => '1',`
	` 'CustomerID' => '2',`
	` 'CustomerName' => '1',`
	` 'CustomerUserID' => '1',`
	` 'EscalationResponseTime' => '1',`
	` 'EscalationSolutionTime' => '1',`
	` 'EscalationTime' => '1',`
	` 'EscalationUpdateTime' => '1',`
	` 'Lock' => '2',`

	'Owner' => '2', 'PendingTime' => '1', 'Priority' => '1', 'Queue' => '2', 'Responsible' => '1', 'SLA' => '1', 'Service' => '1', 'State' => '2', 'TicketNumber' => '2', 'Title' => '2', 'Type' => '1' };

6.52.11. Ticket::Frontend::AgentTicketService###DefaultColumns

Description:	Columns that can be filtered in the service view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	0

Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketService'}->{'DefaultColumns'} =
	{
	'Age' => '2',
	'Changed' => '1',
	'CustomerID' => '2',
	'CustomerName' => '1',
	'CustomerUserID' => '1',
	'EscalationResponseTime' => '1',
	'EscalationSolutionTime' => '1',
	'EscalationTime' => '1',
	'EscalationUpdateTime' => '1',
	'Lock' => '2',
	'Owner' => '2',
	'PendingTime' => '1',
	'Priority' => '1',
	'Queue' => '2',
	'Responsible' => '1',
	'SLA' => '1',
	'Service' => '2',
	'State' => '2',
	'TicketNumber' => '2',
	'Title' => '2',
	'Type' => '1'
	};

6.53. Frontend::Agent::Ticket::ViewResponsible

6.53.1. Ticket::Frontend::AgentTicketResponsibleView###SortBy::Default

Description:	Defines the default ticket attribute for ticket sorting in the responsible view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsibleView'}->{'SortBy::Default'} =  'Age';`

6.53.2. Ticket::Frontend::AgentTicketResponsibleView###Order::Default

Description:	Defines the default ticket order in the responsible view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsibleView'}->{'Order::Default'} =  'Up';`

6.53.3. Ticket::Frontend::AgentTicketResponsible###Permission

Description:	Required permissions to use the ticket responsible screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Permission'} =  'responsible';`

6.53.4. Ticket::Frontend::AgentTicketResponsible###RequiredLock

Description:	Defines if a ticket lock is required in the ticket responsible screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'RequiredLock'} =  '0';`

6.53.5. Ticket::Frontend::AgentTicketResponsible###TicketType

Description:	Sets the ticket type in the ticket responsible screen of the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'TicketType'} =  '0';`

6.53.6. Ticket::Frontend::AgentTicketResponsible###Service

Description:	Sets the service in the ticket responsible screen of the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Service'} = '0';`

6.53.7. Ticket::Frontend::AgentTicketResponsible###ServiceMandatory

Description:	Sets if service must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'ServiceMandatory'} =  '0';`

6.53.8. Ticket::Frontend::AgentTicketResponsible###SLAMandatory

Description:	Sets if SLA must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'SLAMandatory'} =  '0';`

6.53.9. Ticket::Frontend::AgentTicketResponsible###Queue

Description:	Sets the queue in the ticket responsible screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Queue'} = '0';`

6.53.10. Ticket::Frontend::AgentTicketResponsible###Owner

Description:	Sets the ticket owner in the ticket responsible screen of the agent interface.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Owner'} = '0';`

6.53.11. Ticket::Frontend::AgentTicketResponsible###OwnerMandatory

Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'OwnerMandatory'} =  '0';`

6.53.12. Ticket::Frontend::AgentTicketResponsible###Responsible

Description:	Sets the responsible agent of the ticket in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Responsible'} =  '1';`

6.53.13. Ticket::Frontend::AgentTicketResponsible###State

Description:	If a note is added by an agent, sets the state of a ticket in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'State'} = '0';`

6.53.14. Ticket::Frontend::AgentTicketResponsible###StateType

Description:	Defines the next state of a ticket after adding a note, in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible

Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'StateType'} =`
	`[`
	`  'open',`
	`  'pending reminder',`
	`  'pending auto'`
	`];`

6.53.15. Ticket::Frontend::AgentTicketResponsible###StateDefault

Description:	Defines the default next state of a ticket after adding a note, in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'StateDefault'}`
	`=  'open';`

6.53.16. Ticket::Frontend::AgentTicketResponsible###Note

Description:	Allows adding notes in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Note'} =  '1';`

6.53.17. Ticket::Frontend::AgentTicketResponsible###NoteMandatory

Description:	Sets if note must be filled in by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}-`
	`>{'NoteMandatory'} =  '1';`

6.53.18. Ticket::Frontend::AgentTicketResponsible###Subject

Description:	Sets the default subject for notes added in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1

Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Subject'} = '[% Translate("Responsible Update") | html %]!';`

6.53.19. Ticket::Frontend::AgentTicketResponsible###Body

Description:	Sets the default body text for notes added in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Body'} =  '';`

6.53.20. Ticket::Frontend::AgentTicketResponsible###InvolvedAgent

Description:	Shows a list of all the involved agents on this ticket, in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'InvolvedAgent'} =  '0';`

6.53.21. Ticket::Frontend::AgentTicketResponsible###InformAgent

Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'InformAgent'} =  '0';`

6.53.22. Ticket::Frontend::AgentTicketResponsible###ArticleTypeDefault

Description:	Defines the default type of the note in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'ArticleTypeDefault'} =  'note-internal';`

6.53.23. Ticket::Frontend::AgentTicketResponsible###ArticleTypes

Description:	Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'ArticleTypes'} =  {`
	`  'note-external' => '0',`
	`  'note-internal' => '1',`
	`  'note-report' => '0'`
	`};`

6.53.24. Ticket::Frontend::AgentTicketResponsible###Priority

Description:	Shows the ticket priority options in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Priority'} = '0';`

6.53.25. Ticket::Frontend::AgentTicketResponsible###PriorityDefault

Description:	Defines the default ticket priority in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'PriorityDefault'} =  '3 normal';`

6.53.26. Ticket::Frontend::AgentTicketResponsible###Title

Description:	Shows the title fields in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Title'} = '1';`

6.53.27. Ticket::Frontend::AgentTicketResponsible###HistoryType

Description:	Defines the history type for the ticket responsible screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'HistoryType'} =  'AddNote';`

6.53.28. Ticket::Frontend::AgentTicketResponsible###HistoryComment

Description:	Defines the history comment for the ticket responsible screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'HistoryComment'} =  '%%Responsible';`

6.53.29. Ticket::Frontend::AgentTicketResponsible###DynamicField

Description:	Dynamic fields shown in the ticket responsible screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'DynamicField'} =  {};`

6.53.30. Ticket::Frontend::AgentTicketResponsible###RichTextWidth

Description:	Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'RichTextWidth'} =  '620';`

6.53.31. Ticket::Frontend::AgentTicketResponsible###RichTextHeight

Description:	Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'RichTextHeight'} =  '100';`

6.53.32. Ticket::Frontend::AgentTicketResponsibleView###DefaultColumns

Description:	Columns that can be filtered in the responsible view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsibleView'}->{'DefaultColumns'} =  {`
	`  'Age' => '2',`
	`  'Changed' => '1',`
	`  'CustomerID' => '2',`
	`  'CustomerName' => '1',`
	`  'CustomerUserID' => '1',`
	`  'EscalationResponseTime' => '1',`
	`  'EscalationSolutionTime' => '1',`
	`  'EscalationTime' => '1',`
	`  'EscalationUpdateTime' => '1',`
	`  'Lock' => '2',`
	`  'Owner' => '2',`
	`  'PendingTime' => '1',`
	`  'Priority' => '1',`
	`  'Queue' => '2',`
	`  'Responsible' => '1',`
	`  'SLA' => '1',`
	`  'Service' => '1',`
	`  'State' => '2',`
	`  'TicketNumber' => '2',`
	`  'Title' => '2',`
	`  'Type' => '1'`
	`};`

6.54. Frontend::Agent::Ticket::ViewSearch

6.54.1. Ticket::Frontend::AgentTicketSearch###ExtendedSearchCondition

Description:	Allows extended search conditions in ticket search of the agent interface. With this feature you can search e. 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:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'ExtendedSearchCondition'} =  '1';`

6.54.2. Ticket::Frontend::AgentTicketSearch###SearchLimit

Description:	Maximum number of tickets to be displayed in the result of a search in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchLimit'} = '2000';`

6.54.3. Ticket::Frontend::AgentTicketSearch###SearchPageShown

Description:	Number of tickets to be displayed in each page of a search result in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchPageShown'} =  '40';`

6.54.4. Ticket::Frontend::AgentTicketSearch###SearchViewableTicketLines

Description:	Number of lines (per ticket) that are shown by the search utility in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchViewableTicketLines'} =  '10';`

6.54.5. Ticket::Frontend::AgentTicketSearch###SortBy::Default

Description:	Defines the default ticket attribute for ticket sorting of the ticket search result of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1

Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SortBy::Default'} = 'Age';

6.54.6. Ticket::Frontend::AgentTicketSearch###Order::Default

Description:	Defines the default ticket order in the ticket search result of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Order::Default'} = 'Down';

6.54.7. Ticket::Frontend::AgentTicketSearch###SearchArticleCSVTree

Description:	Exports the whole article tree in search result (it can affect the system performance).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchArticleCSVTree'} = '0';

6.54.8. Ticket::Frontend::AgentTicketSearch###SearchCSVData

Description:	Data used to export the search result in CSV format.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchCSVData'} = ['TicketNumber', 'Age', 'Created', 'Closed', 'FirstLock', 'FirstResponse', 'State', 'Priority', 'Queue', 'Lock', 'Owner', 'UserFirstname', 'UserLastname', 'CustomerID', 'CustomerName', 'From', 'Subject', 'AccountedTime', 'ArticleTree',

	`  'SolutionInMin',`
	`  'SolutionDiffInMin',`
	`  'FirstResponseInMin',`
	`  'FirstResponseDiffInMin'`
	`];`

6.54.9. Ticket::Frontend::AgentTicketSearch###ArticleCreateTime

Description:	Includes article create times in the ticket search of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'ArticleCreateTime'} =  '0';`

6.54.10. Ticket::Frontend::AgentTicketSearch###Defaults###Fulltext

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Fulltext'} =  '';`

6.54.11. Ticket::Frontend::AgentTicketSearch###Defaults###TicketNumber

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketNumber'} =  '';`

6.54.12. Ticket::Frontend::AgentTicketSearch###Defaults###Title

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Title'} =  '';`

6.54.13. Ticket::Frontend::AgentTicketSearch###Defaults###From

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'From'} =  '';`

6.54.14. Ticket::Frontend::AgentTicketSearch###Defaults###To

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'To'} =  '';`

6.54.15. Ticket::Frontend::AgentTicketSearch###Defaults###Cc

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Cc'} =  '';`

6.54.16. Ticket::Frontend::AgentTicketSearch###Defaults###Subject

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Subject'} =  '';`

6.54.17. Ticket::Frontend::AgentTicketSearch###Defaults###Body

Description:	Defines the default shown ticket search attribute for ticket search screen.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Body'} =  '';`

6.54.18. Ticket::Frontend::AgentTicketSearch###Defaults###CustomerID

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'CustomerID'} =  '';`

6.54.19. Ticket::Frontend::AgentTicketSearch###Defaults###CustomerUserLogin

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'CustomerUserLogin'} =  '';`

6.54.20. Ticket::Frontend::AgentTicketSearch###Defaults###StateIDs

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'StateIDs'} =  [];`

6.54.21. Ticket::Frontend::AgentTicketSearch###Defaults###QueueIDs

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch

Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'QueueIDs'} =  [];`

6.54.22. Ticket::Frontend::AgentTicketSearch###Defaults###TicketCreateTimePoint

Description:	Default data to use on attribute for ticket search screen. Example: "TicketCreateTimePointFormat=year;TicketCreateTimePointStart=Last;TicketCreate
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCreateTimePoint'} =  '';`

6.54.23. Ticket::Frontend::AgentTicketSearch###Defaults###TicketCreateTimeSlot

Description:	Default data to use on attribute for ticket search screen. Example: "TicketCreateTimeStartYear=2010;TicketCreateTimeStartMonth=10;TicketCreateTi
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCreateTimeSlot'} =  '';`

6.54.24. Ticket::Frontend::AgentTicketSearch###Defaults###TicketChangeTimePoint

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketChangeTimePoint'} =  '';`

6.54.25. Ticket::Frontend::AgentTicketSearch###Defaults###TicketChangeTimeSlot

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch

Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketChangeTimeSlot'} =  '';`

6.54.26. Ticket::Frontend::AgentTicketSearch###Defaults###TicketCloseTimePoint

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCloseTimePoint'} =  '';`

6.54.27. Ticket::Frontend::AgentTicketSearch###Defaults###TicketCloseTimeSlot

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCloseTimeSlot'} =  '';`

6.54.28. Ticket::Frontend::AgentTicketSearch###Defaults###TicketEscalationTimePoint

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketEscalationTimePoint'} =  '';`

6.54.29. Ticket::Frontend::AgentTicketSearch###Defaults###TicketEscalationTimeSlot

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch

Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketEscalationTimeSlot'} =  '';`

6.54.30. Ticket::Frontend::AgentTicketSearch###Defaults###ArticleCreateTimePoint

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'ArticleCreateTimePoint'} =  '';`

6.54.31. Ticket::Frontend::AgentTicketSearch###Defaults###ArticleCreateTimeSlot

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'ArticleCreateTimeSlot'} =  '';`

6.54.32. Ticket::Frontend::AgentTicketSearch###Defaults###SearchInArchive

Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'SearchInArchive'} =  '';`

6.54.33. Ticket::Frontend::CustomerTicketSearch###SearchArticleCSVTree

Description:	Exports the whole article tree in search result (it can affect the system performance).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchArticleCSVTree'} =  '0';`

6.54.34. Ticket::Frontend::AgentTicketSearch###DynamicField

Description:	Dynamic fields shown in the ticket search screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and shown by default.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'DynamicField'} = {};`

6.54.35. Ticket::Frontend::AgentTicketSearch###Defaults###DynamicField

Description:	Defines the default shown ticket search attribute for ticket search screen. Example: "Key" must have the name of the Dynamic Field in this case 'X', "Content" must have the value of the Dynamic Field depending on the Dynamic Field type, Text: 'a text', Dropdown: '1', Date/Time: 'Search_DynamicField_XTimeSlotStartYear=1974; Search_DynamicField_XTimeSlotStartMonth=01; Search_DynamicField_XTimeSlotStartDay=26; Search_DynamicField_XTimeSlotStartHour=00; Search_DynamicField_XTimeSlotStartMinute=00; Search_DynamicField_XTimeSlotStartSecond=00; Search_DynamicField_XTimeSlotStopYear=2013; Search_DynamicField_XTimeSlotStopMonth=01; Search_DynamicField_XTimeSlotStopDay=26; Search_DynamicField_XTimeSlotStopHour=23; Search_DynamicField_XTimeSlotStopMinute=59; Search_DynamicField_XTimeSlotStopSecond=59;' and or 'Search_DynamicField_XTimePointFormat=week; Search_DynamicField_XTimePointStart=Before; Search_DynamicField_XTimePointValue=7';.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'DynamicField'} =  {};`

6.54.36. Ticket::Frontend::AgentTicketSearch###SearchCSVDynamicField

Description:	Dynamic Fields used to export the search result in CSV format.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch

Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchCSVDynamicField'} =  {};`

6.54.37. Ticket::Frontend::AgentTicketSearch###DefaultColumns

Description:	Columns that can be filtered in the ticket search result view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'DefaultColumns'} =`
	`{`
	`'Age' => '2',`
	`'Changed' => '1',`
	`'CustomerID' => '2',`
	`'CustomerName' => '1',`
	`'CustomerUserID' => '1',`
	`'EscalationResponseTime' => '1',`
	`'EscalationSolutionTime' => '1',`
	`'EscalationTime' => '1',`
	`'EscalationUpdateTime' => '1',`
	`'Lock' => '2',`
	`'Owner' => '2',`
	`'PendingTime' => '1',`
	`'Priority' => '1',`
	`'Queue' => '2',`
	`'Responsible' => '1',`
	`'SLA' => '1',`
	`'Service' => '1',`
	`'State' => '2',`
	`'TicketNumber' => '2',`
	`'Title' => '2',`
	`'Type' => '1'`
	`};`

6.55. Frontend::Agent::Ticket::ViewService

6.55.1. Ticket::Frontend::AgentTicketService###StripEmptyLines

Description:	Strips empty lines on the ticket preview in the service view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewService
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketService'}->{'StripEmptyLines'} =  '0';`

6.55.2. Ticket::Frontend::AgentTicketService###ViewAllPossibleTickets

Description:	Shows all both ro and rw tickets in the service view.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewService
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketService'}-`
	`>{'ViewAllPossibleTickets'} =  '0';`

6.55.3. Ticket::Frontend::AgentTicketService###ServiceSort

Description:	Sorts the tickets (ascendingly or descendingly) when a single queue is selected in the service view and after the tickets are sorted by priority. Values: 0 = ascending (oldest on top, default), 1 = descending (youngest on top). Use the ServiceID for the key and 0 or 1 for value.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewService
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketService'}->{'ServiceSort'} =  {`
	`  '3' => '0',`
	`  '7' => '1'`
	`};`

6.55.4. Ticket::Frontend::AgentTicketService###SortBy::Default

Description:	Defines the default sort criteria for all services displayed in the service view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewService
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketService'}->{'SortBy::Default'}`
	`=  'Age';`

6.55.5. Ticket::Frontend::AgentTicketService###PreSort::ByPriority

Description:	Defines if a pre-sorting by priority should be done in the service view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewService
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketService'}-`
	`>{'PreSort::ByPriority'} =  '1';`

6.55.6. Ticket::Frontend::AgentTicketService###Order::Default

Description:	Defines the default sort order for all services in the service view, after priority sort.

Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewService
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketService'}->{'Order::Default'} = 'Up';`

6.56. Frontend::Agent::Ticket::ViewStatus

6.56.1. Ticket::Frontend::AgentTicketStatusView###ViewableTicketsPage

Description:	Shows all open tickets (even if they are locked) in the status view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'ViewableTicketsPage'} = '50';`

6.56.2. Ticket::Frontend::AgentTicketStatusView###SortBy::Default

Description:	Defines the default ticket attribute for ticket sorting in the status view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'SortBy::Default'} = 'Age';`

6.56.3. Ticket::Frontend::AgentTicketStatusView###Order::Default

Description:	Defines the default ticket order (after priority sort) in the status view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'Order::Default'} = 'Down';`

6.56.4. Ticket::Frontend::AgentTicketStatusView###DefaultColumns

Description:	Columns that can be filtered in the status view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 =

	Enabled by default. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'DefaultColumns'} =  {`
	`  'Age' => '2',`
	`  'Changed' => '1',`
	`  'CustomerID' => '2',`
	`  'CustomerName' => '1',`
	`  'CustomerUserID' => '1',`
	`  'EscalationResponseTime' => '1',`
	`  'EscalationSolutionTime' => '1',`
	`  'EscalationTime' => '1',`
	`  'EscalationUpdateTime' => '1',`
	`  'Lock' => '2',`
	`  'Owner' => '2',`
	`  'PendingTime' => '1',`
	`  'Priority' => '1',`
	`  'Queue' => '2',`
	`  'Responsible' => '1',`
	`  'SLA' => '1',`
	`  'Service' => '1',`
	`  'State' => '2',`
	`  'TicketNumber' => '2',`
	`  'Title' => '2',`
	`  'Type' => '1'`
	`};`

6.57. Frontend::Agent::Ticket::ViewWatch

6.57.1. Ticket::Frontend::AgentTicketWatchView###SortBy::Default

Description:	Defines the default ticket attribute for ticket sorting in the watch view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewWatch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketWatchView'}->{'SortBy::Default'} =  'Age';`

6.57.2. Ticket::Frontend::AgentTicketWatchView###Order::Default

Description:	Defines the default ticket order in the watch view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewWatch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketWatchView'}->{'Order::Default'} =  'Up';`

6.57.3. Ticket::Frontend::AgentTicketWatchView###DefaultColumns

Description:	Columns that can be filtered in the watch view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewWatch
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketWatchView'}->{'DefaultColumns'} =  {`
	`  'Age' => '2',`
	`  'Changed' => '1',`
	`  'CustomerID' => '2',`
	`  'CustomerName' => '1',`
	`  'CustomerUserID' => '1',`
	`  'EscalationResponseTime' => '1',`
	`  'EscalationSolutionTime' => '1',`
	`  'EscalationTime' => '1',`
	`  'EscalationUpdateTime' => '1',`
	`  'Lock' => '2',`
	`  'Owner' => '2',`
	`  'PendingTime' => '1',`
	`  'Priority' => '1',`
	`  'Queue' => '2',`
	`  'Responsible' => '1',`
	`  'SLA' => '1',`
	`  'Service' => '1',`
	`  'State' => '2',`
	`  'TicketNumber' => '2',`
	`  'Title' => '2',`
	`  'Type' => '1'`
	`};`

6.58. Frontend::Agent::Ticket::ViewZoom

6.58.1. Ticket::Frontend::PlainView

Description:	Shows a link to see a zoomed email ticket in plain text.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::PlainView'} =  '0';`

6.58.2. Ticket::Frontend::ZoomExpand

Description:	Shows all the articles of the ticket (expanded) in the zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::ZoomExpand'} =  '0';`

6.58.3. Ticket::Frontend::ZoomExpandSort

Description:	Shows the articles sorted normally or in reverse, under ticket zoom in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ZoomExpandSort'} =  'normal';`

6.58.4. Ticket::ZoomAttachmentDisplayCount

Description:	Shows a count of icons in the ticket zoom, if the article has attachments.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ZoomAttachmentDisplayCount'} =  '20';`

6.58.5. Ticket::ZoomTimeDisplay

Description:	Displays the accounted time for an article in the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ZoomTimeDisplay'} =  '0';`

6.58.6. Ticket::UseArticleColors

Description:	Shows colors for different article types in the article table.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::UseArticleColors'} =  '0';`

6.58.7. Ticket::Frontend::TicketArticleFilter

Description:	Activates the article filter in the zoom view to specify which articles should be shown.
Group:	Ticket

SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::TicketArticleFilter'} =  '0';`

6.58.8. Ticket::Frontend::HTMLArticleHeightDefault

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:	`$Self->{'Ticket::Frontend::HTMLArticleHeightDefault'} =  '100';`

6.58.9. Ticket::Frontend::HTMLArticleHeightMax

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:	`$Self->{'Ticket::Frontend::HTMLArticleHeightMax'} =  '2500';`

6.58.10. Ticket::Frontend::MaxArticlesZoomExpand

Description:	The maximal number of articles expanded on a single page in AgentTicketZoom.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::MaxArticlesZoomExpand'} =  '400';`

6.58.11. Ticket::Frontend::MaxArticlesPerPage

Description:	The maximal number of articles shown on a single page in AgentTicketZoom.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::MaxArticlesPerPage'} =  '1000';`

6.58.12. Ticket::Frontend::ZoomRichTextForce

Description:	Show article as rich text even if rich text writing is disabled.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ZoomRichTextForce'} =  '1';`

6.58.13. Ticket::Frontend::AgentTicketZoom###DynamicField

Description:	Dynamic fields shown in the sidebar of the ticket zoom screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketZoom'}->{'DynamicField'} =  {};`

6.59. Frontend::Agent::TicketOverview

6.59.1. Ticket::Frontend::Overview###Small

Description:	Allows having a small format ticket overview (CustomerInfo => 1 - shows also the customer information).
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::Overview'}->{'Small'} =  {`
	`  'CustomerInfo' => '1',`
	`  'Module' => 'Kernel::Output::HTML::TicketOverviewSmall',`
	`  'ModulePriority' => '100',`
	`  'Name' => 'Small',`
	`  'NameShort' => 'S'`
	`};`

6.59.2. Ticket::Frontend::OverviewSmall###ColumnHeader

Description:	Shows either the last customer article's subject or the ticket title in the small format overview.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::OverviewSmall'}->{'ColumnHeader'} =`
	`'LastCustomerSubject';`

6.59.3. Ticket::Frontend::Overview###Medium

Description:	Allows having a medium format ticket overview (CustomerInfo => 1 - shows also the customer information).
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::Overview'}->{'Medium'} =  {`
	`  'CustomerInfo' => '0',`
	`  'Module' => 'Kernel::Output::HTML::TicketOverviewMedium',`
	`  'ModulePriority' => '200',`
	`  'Name' => 'Medium',`
	`  'NameShort' => 'M',`
	`  'OverviewMenuModules' => '1',`
	`  'TicketActionsPerTicket' => '0'`
	`};`

6.59.4. Ticket::Frontend::Overview###Preview

Description:	Shows a preview of the ticket overview (CustomerInfo => 1 - shows also Customer-Info, CustomerInfoMaxSize max. size in characters of Customer-Info).
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::Overview'}->{'Preview'} =  {`
	`  'CustomerInfo' => '0',`
	`  'CustomerInfoMaxSize' => '18',`
	`  'DefaultPreViewLines' => '25',`
	`  'DefaultViewNewLine' => '90',`
	`  'Module' => 'Kernel::Output::HTML::TicketOverviewPreview',`
	`  'ModulePriority' => '300',`
	`  'Name' => 'Preview',`
	`  'NameShort' => 'L',`
	`  'OverviewMenuModules' => '1',`
	`  'StripEmptyLines' => '0',`
	`  'TicketActionsPerTicket' => '0'`
	`};`

6.59.5. Ticket::Frontend::Overview::PreviewArticleSenderTypes

Description:	Defines which article sender types should be shown in the preview of a ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::Overview::PreviewArticleSenderTypes'} =  {`
	`  'agent' => '1',`
	`  'customer' => '1',`
	`  'system' => '1'`

	`};`

6.59.6. Ticket::Frontend::Overview::PreviewArticleLimit

Description:	Sets the count of articles visible in preview mode of ticket overviews.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::Overview::PreviewArticleLimit'} =  '5';`

6.59.7. Ticket::Frontend::Overview::PreviewArticleTypeExpanded

Description:	Defines wich article type should be expanded when entering the overview. If nothing defined, latest article will be expanded.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::Overview::PreviewArticleTypeExpanded'} = '';`

6.59.8. Ticket::Frontend::OverviewSmall###DynamicField

Description:	Dynamic fields shown in the ticket small format overview screen of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::OverviewSmall'}->{'DynamicField'} =  {};`

6.59.9. Ticket::Frontend::OverviewMedium###DynamicField

Description:	Dynamic fields shown in the ticket medium format overview screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::OverviewMedium'}->{'DynamicField'} =  {};`

6.59.10. Ticket::Frontend::OverviewPreview###DynamicField

Description:	Dynamic fields shown in the ticket preview format overview screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.

Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::OverviewPreview'}->{'DynamicField'} =  {};`

6.60. Frontend::Agent::ToolBarModule

6.60.1. Frontend::ToolBarModule###1-Ticket::AgentTicketQueue

Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'1-Ticket::AgentTicketQueue'} =`
	`{`
	`'AccessKey' => 'q',`
	`'Action' => 'AgentTicketQueue',`
	`'CssClass' => 'QueueView',`
	`'Icon' => 'fa fa-folder',`
	`'Link' => 'Action=AgentTicketQueue',`
	`'Module' => 'Kernel::Output::HTML::ToolBarLink',`
	`'Name' => 'Queue view',`
	`'Priority' => '1010010'`
	`};`

6.60.2. Frontend::ToolBarModule###2-Ticket::AgentTicketStatus

Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'2-Ticket::AgentTicketStatus'} =`
	`{`
	`'AccessKey' => 'S',`
	`'Action' => 'AgentTicketStatusView',`
	`'CssClass' => 'StatusView',`
	`'Icon' => 'fa fa-list-ol',`
	`'Link' => 'Action=AgentTicketStatusView',`
	`'Module' => 'Kernel::Output::HTML::ToolBarLink',`
	`'Name' => 'Status view',`
	`'Priority' => '1010020'`
	`};`

6.60.3. Frontend::ToolBarModule###3-Ticket::AgentTicketEscalation

Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule

Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'3-Ticket::AgentTicketEscalation'} =  {`
	`  'AccessKey' => 'w',`
	`  'Action' => 'AgentTicketEscalationView',`
	`  'CssClass' => 'EscalationView',`
	`  'Icon' => 'fa fa-exclamation',`
	`  'Link' => 'Action=AgentTicketEscalationView',`
	`  'Module' => 'Kernel::Output::HTML::ToolBarLink',`
	`  'Name' => 'Escalation view',`
	`  'Priority' => '1010030'`
	`};`

6.60.4. Frontend::ToolBarModule###4-Ticket::AgentTicketPhone

Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'4-Ticket::AgentTicketPhone'} =`
	`{`
	`  'AccessKey' => '',`
	`  'Action' => 'AgentTicketPhone',`
	`  'CssClass' => 'PhoneTicket',`
	`  'Icon' => 'fa fa-phone',`
	`  'Link' => 'Action=AgentTicketPhone',`
	`  'Module' => 'Kernel::Output::HTML::ToolBarLink',`
	`  'Name' => 'New phone ticket',`
	`  'Priority' => '1020010'`
	`};`

6.60.5. Frontend::ToolBarModule###5-Ticket::AgentTicketEmail

Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'5-Ticket::AgentTicketEmail'} =`
	`{`
	`  'AccessKey' => '',`
	`  'Action' => 'AgentTicketEmail',`
	`  'CssClass' => 'EmailTicket',`
	`  'Icon' => 'fa fa-envelope',`
	`  'Link' => 'Action=AgentTicketEmail',`
	`  'Module' => 'Kernel::Output::HTML::ToolBarLink',`
	`  'Name' => 'New email ticket',`
	`  'Priority' => '1020020'`
	`};`

6.60.6. Frontend::ToolBarModule###6-Ticket::AgentTicketProcess

Description:	Toolbar Item for a shortcut.

Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	$Self->{'Frontend::ToolBarModule'}->{'6-Ticket::AgentTicketProcess'}
	= {
	  'AccessKey' => 'p',
	  'Action' => 'AgentTicketProcess',
	  'CssClass' => 'ProcessTicket',
	  'Icon' => 'fa fa-th-large',
	  'Link' => 'Action=AgentTicketProcess',
	  'Module' => 'Kernel::Output::HTML::ToolBarLink',
	  'Name' => 'New process ticket',
	  'Priority' => '1020030'
	};

6.60.7. Frontend::ToolBarModule###6-Ticket::TicketResponsible

Description:	Agent interface notification module to see the number of tickets an agent is responsible for.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::ToolBarModule'}->{'6-Ticket::TicketResponsible'} =
	  {
	  'AccessKey' => 'r',
	  'AccessKeyNew' => '',
	  'AccessKeyReached' => '',
	  'CssClass' => 'Responsible',
	  'CssClassNew' => 'Responsible New',
	  'CssClassReached' => 'Responsible Reached',
	  'Icon' => 'fa fa-user',
	  'IconNew' => 'fa fa-user',
	  'IconReached' => 'fa fa-user',
	  'Module' => 'Kernel::Output::HTML::ToolBarTicketResponsible',
	  'Priority' => '1030010'
	};

6.60.8. Frontend::ToolBarModule###7-Ticket::TicketWatcher

Description:	Agent interface notification module to see the number of watched tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::ToolBarModule'}->{'7-Ticket::TicketWatcher'} = {
	  'AccessKey' => '',
	  'AccessKeyNew' => '',
	  'AccessKeyReached' => '',
	  'CssClass' => 'Watcher',
	  'CssClassNew' => 'Watcher New',
	  'CssClassReached' => 'Watcher Reached',

	'Icon' => 'fa fa-eye', 'IconNew' => 'fa fa-eye', 'IconReached' => 'fa fa-eye', 'Module' => 'Kernel::Output::HTML::ToolBarTicketWatcher', 'Priority' => '1030020' };

6.60.9. Frontend::ToolBarModule###8-Ticket::TicketLocked

Description:	Agent interface notification module to check the used charset.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	1
Required:	0

Config-Setting:	$Self->{'Frontend::ToolBarModule'}->{'8-Ticket::TicketLocked'} = {
	'AccessKey' => 'k',
	'AccessKeyNew' => '',
	'AccessKeyReached' => '',
	'CssClass' => 'Locked',
	'CssClassNew' => 'Locked New',
	'CssClassReached' => 'Locked Reached',
	'Icon' => 'fa fa-lock',
	'IconNew' => 'fa fa-lock',
	'IconReached' => 'fa fa-lock',
	'Module' => 'Kernel::Output::HTML::ToolBarTicketLocked',
	'Priority' => '1030030'
	};

6.60.10. Frontend::ToolBarModule###8-Ticket::AgentTicketService

Description:	Agent interface notification module to see the number of tickets in My Services.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	$Self->{'Frontend::ToolBarModule'}->{'8-Ticket::AgentTicketService'}
	= {
	'CssClass' => 'ServiceView',
	'Icon' => 'fa fa-wrench',
	'Module' => 'Kernel::Output::HTML::ToolBarTicketService',
	'Priority' => '1030035'
	};

6.60.11. Frontend::ToolBarModule###9-Ticket::TicketSearchProfile

Description:	Agent interface module to access search profiles via nav bar.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0
Config-Setting:	$Self->{'Frontend::ToolBarModule'}->{'9-Ticket::TicketSearchProfile'} = {

	'Block' => 'ToolBarSearchProfile',
	  'Description' => 'Search-Template',
	  'MaxWidth' => '40',
	  'Module' => 'Kernel::Output::HTML::ToolBarTicketSearchProfile',
	  'Name' => 'Search-Template',
	  'Priority' => '1990010'
	};

6.60.12. Frontend::ToolBarModule###10-Ticket::TicketSearchFulltext

Description:	Agent interface module to access fulltext search via nav bar.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	$Self->{'Frontend::ToolBarModule'}->{'10-Ticket::TicketSearchFulltext'} = {
	  'Block' => 'ToolBarSearchFulltext',
	  'CSS' => 'Core.Agent.Toolbar.FulltextSearch.css',
	  'Description' => 'Fulltext search',
	  'Module' => 'Kernel::Output::HTML::ToolBarGeneric',
	  'Name' => 'Fulltext-Search',
	  'Priority' => '1990020',
	  'Size' => '10'
	};

6.60.13. Frontend::ToolBarModule###11-CICSearchCustomerID

Description:	Agent interface module to access CIC search via nav bar.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	$Self->{'Frontend::ToolBarModule'}->{'11-CICSearchCustomerID'} = {
	  'Block' => 'ToolBarCICSearchCustomerID',
	  'CSS' => 'Core.Agent.Toolbar.CICSearch.css',
	  'Description' => 'CustomerID Search',
	  'Module' => 'Kernel::Output::HTML::ToolBarGeneric',
	  'Name' => 'CustomerID search',
	  'Priority' => '1990030',
	  'Size' => '10'
	};

6.60.14. Frontend::ToolBarModule###11-CICSearchCustomerUser

Description:	Agent interface module to access CIC search via nav bar.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	$Self->{'Frontend::ToolBarModule'}->{'11-CICSearchCustomerUser'} = {
	  'Block' => 'ToolBarCICSearchCustomerUser',
	  'CSS' => 'Core.Agent.Toolbar.CICSearch.css',

	'Description' => 'CustomerUser Search', 'Module' => 'Kernel::Output::HTML::ToolBarGeneric', 'Name' => 'Customer user search', 'Priority' => '1990040', 'Size' => '10' };

6.61. Frontend::Customer

6.61.1. Ticket::Frontend::CustomerDisableCompanyTicketAccess

Description:	This option will deny the access to customer company tickets, which are not created by the customer user.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	$Self->{'Ticket::Frontend::CustomerDisableCompanyTicketAccess'} = '0';

6.61.2. Ticket::Frontend::CustomerTicketOverviewCustomEmptyText

Description:	Custom text for the page shown to customers that have no tickets yet (if you need those text translated add them to a custom translation module).
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	0
Required:	0

Config-Setting:	$Self->{'Ticket::Frontend::CustomerTicketOverviewCustomEmptyText'} = {
	'Button' => 'Create your first ticket',
	'Text' => 'Please click the button below to create your first ticket.',
	'Title' => 'Welcome!'
	};

6.61.3. Frontend::CustomerUser::Item###15-OpenTickets

Description:	Customer item (icon) which shows the open tickets of this customer as info block. Setting CustomerUserLogin to 1 searches for tickets based on login name rather than CustomerID.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	0
Config-Setting:	$Self->{'Frontend::CustomerUser::Item'}->{'15-OpenTickets'} = { 'Action' => 'AgentTicketSearch', 'Attributes' => 'StateType=Open;', 'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css', 'CSSClassNoOpenTicket' => 'NoOpenTicket', 'CSSClassOpenTicket' => 'OpenTicket',

	'CustomerUserLogin' => '0', 'IconNameNoOpenTicket' => 'fa-check-circle', 'IconNameOpenTicket' => 'fa-exclamation-circle', 'Module' => 'Kernel::Output::HTML::CustomerUserGenericTicket', 'Subaction' => 'Search', 'Target' => '_blank', 'Text' => 'Open tickets (customer)' };

6.61.4. Frontend::CustomerUser::Item###16-OpenTicketsForCustomerUserLogin

Description:	Customer item (icon) which shows the open tickets of this customer as info block. Setting CustomerUserLogin to 1 searches for tickets based on login name rather than CustomerID.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	0
Required:	0

Config-Setting:	$Self->{'Frontend::CustomerUser::Item'}->{'16-OpenTicketsForCustomerUserLogin'} = {
	'Action' => 'AgentTicketSearch',
	'Attributes' => 'StateType=Open;',
	'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css',
	'CSSClassNoOpenTicket' => 'NoOpenTicket',
	'CSSClassOpenTicket' => 'OpenTicket',
	'CustomerUserLogin' => '1',
	'IconNameNoOpenTicket' => 'fa-check-circle',
	'IconNameOpenTicket' => 'fa-exclamation-circle',
	'Module' => 'Kernel::Output::HTML::CustomerUserGenericTicket',
	'Subaction' => 'Search',
	'Target' => '_blank',
	'Text' => 'Open tickets (customer user)'
	};

6.61.5. Frontend::CustomerUser::Item###17-ClosedTickets

Description:	Customer item (icon) which shows the closed tickets of this customer as info block. Setting CustomerUserLogin to 1 searches for tickets based on login name rather than CustomerID.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	0
Required:	0

Config-Setting:	$Self->{'Frontend::CustomerUser::Item'}->{'17-ClosedTickets'} = {
	'Action' => 'AgentTicketSearch',
	'Attributes' => 'StateType=Closed;',
	'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css',
	'CSSClassNoOpenTicket' => 'NoOpenTicket',
	'CSSClassOpenTicket' => 'OpenTicket',
	'CustomerUserLogin' => '0',
	'IconNameNoOpenTicket' => 'fa-power-off',
	'IconNameOpenTicket' => 'fa-power-off',
	'Module' => 'Kernel::Output::HTML::CustomerUserGenericTicket',
	'Subaction' => 'Search',
	'Target' => '_blank',
	'Text' => 'Closed tickets (customer)'
	};

6.61.6. Frontend::CustomerUser::Item###18-ClosedTicketsForCustomerUserLogin

Description:	Customer item (icon) which shows the closed tickets of this customer as info block. Setting CustomerUserLogin to 1 searches for tickets based on login name rather than CustomerID.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::CustomerUser::Item'}->{'18-ClosedTicketsForCustomerUserLogin'} =  {`
	`  'Action' => 'AgentTicketSearch',`
	`  'Attributes' => 'StateType=Closed;',`
	`  'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css',`
	`  'CSSClassNoOpenTicket' => 'NoOpenTicket',`
	`  'CSSClassOpenTicket' => 'OpenTicket',`
	`  'CustomerUserLogin' => '1',`
	`  'IconNameNoOpenTicket' => 'fa-power-off',`
	`  'IconNameOpenTicket' => 'fa-power-off',`
	`  'Module' => 'Kernel::Output::HTML::CustomerUserGenericTicket',`
	`  'Subaction' => 'Search',`
	`  'Target' => '_blank',`
	`  'Text' => 'Closed tickets (customer user)'`
	`};`

6.61.7. CustomerFrontend::CommonObject###QueueObject

Description:	Path of the file that stores all the settings for the QueueObject object for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerFrontend::CommonObject'}->{'QueueObject'} = 'Kernel::System::Queue';`

6.61.8. CustomerFrontend::CommonObject###TicketObject

Description:	Path of the file that stores all the settings for the TicketObject for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerFrontend::CommonObject'}->{'TicketObject'} = 'Kernel::System::Ticket';`

6.61.9. CustomerFrontend::CommonParam###Action

Description:	Defines the default used Frontend-Module if no Action parameter given in the url on the customer interface.

Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerFrontend::CommonParam'}->{'Action'} =`
	`'CustomerTicketOverview';`

6.61.10. CustomerFrontend::CommonParam###TicketID

Description:	Default ticket ID used by the system in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerFrontend::CommonParam'}->{'TicketID'} =  '';`

6.62. Frontend::Customer::ModuleMetaHead

6.62.1. CustomerFrontend::HeaderMetaModule###2-TicketSearch

Description:	Module to generate html OpenSearch profile for short ticket search in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleMetaHead
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerFrontend::HeaderMetaModule'}->{'2-TicketSearch'} =`
	`{`
	`  'Action' => 'CustomerTicketSearch',`
	`  'Module' => 'Kernel::Output::HTML::CustomerHeaderMetaTicketSearch'`
	`};`

6.63. Frontend::Customer::ModuleRegistration

6.63.1. CustomerFrontend::Module###CustomerTicketOverview

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

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerTicketOverview'} =  {`
	`  'Description' => 'Overview of customer tickets',`
	`  'NavBar' => [`
	`    {`
	`      'AccessKey' => 'm',`
	`      'Block' => '',`
	`      'Description' => 'Tickets',`
	`      'Link' => 'Action=CustomerTicketOverview;Subaction=MyTickets',`
	`      'LinkOption' => '',`
	`      'Name' => 'Tickets',`

```
        'NavBar' => 'Ticket',
        'Prio' => '100',
        'Type' => 'Menu'
      },
      {
        'AccessKey' => '',
        'Block' => '',
        'Description' => 'My Tickets',
        'Link' => 'Action=CustomerTicketOverview;Subaction=MyTickets',
        'LinkOption' => '',
        'Name' => 'My Tickets',
        'NavBar' => 'Ticket',
        'Prio' => '110',
        'Type' => 'Submenu'
      },
      {
        'AccessKey' => 'c',
        'Block' => '',
        'Description' => 'Company Tickets',
        'Link' =>
 'Action=CustomerTicketOverview;Subaction=CompanyTickets',
        'LinkOption' => '',
        'Name' => 'Company Tickets',
        'NavBar' => 'Ticket',
        'Prio' => '120',
        'Type' => 'Submenu'
      }
    ],
    'NavBarName' => 'Ticket',
    'Title' => 'Overview'
};
```

6.63.2. CustomerFrontend::Module###CustomerTicketMessage

Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
 $Self->{'CustomerFrontend::Module'}->{'CustomerTicketMessage'} =  {
    'Description' => 'Create tickets',
    'NavBar' => [
      {
        'AccessKey' => 'n',
        'Block' => '',
        'Description' => 'Create new Ticket',
        'Link' => 'Action=CustomerTicketMessage',
        'LinkOption' => '',
        'Name' => 'New Ticket',
        'NavBar' => 'Ticket',
        'Prio' => '100',
        'Type' => 'Submenu'
      }
    ],
    'NavBarName' => 'Ticket',
    'Title' => 'New Ticket'
};
```

6.63.3. CustomerFrontend::Module###CustomerTicketZoom

Description:	Frontend module registration for the customer interface.

Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerTicketZoom'} =  {`
	`  'Description' => 'Ticket zoom view',`
	`  'Loader' => {`
	`    'JavaScript' => [`
	`      'Core.Customer.TicketZoom.js',`
	`      'Core.UI.Popup.js'`
	`    ]`
	`  },`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'Zoom'`
	`};`

6.63.4. CustomerFrontend::Module###CustomerTicketPrint

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

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerTicketPrint'} =  {`
	`  'Description' => 'Customer Ticket Print Module',`
	`  'NavBarName' => '',`
	`  'Title' => 'Print'`
	`};`

6.63.5. CustomerFrontend::Module###CustomerTicketAttachment

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

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerTicketAttachment'} =`
	`{`
	`  'Description' => 'To download attachments',`
	`  'NavBarName' => '',`
	`  'Title' => ''`
	`};`

6.63.6. CustomerFrontend::Module###CustomerTicketSearch

Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerTicketSearch'} =  {`

<table>
<tr><td></td><td><pre>
  'Description' => 'Customer ticket search',
  'NavBar' => [
    {
      'AccessKey' => 's',
      'Block' => '',
      'Description' => 'Search',
      'Link' => 'Action=CustomerTicketSearch',
      'LinkOption' => '',
      'Name' => 'Search',
      'NavBar' => 'Ticket',
      'Prio' => '300',
      'Type' => 'Submenu'
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'Search'
};
</pre></td></tr>
</table>

6.64. Frontend::Customer::Preferences

6.64.1. CustomerPreferencesGroups###ShownTickets

<table>
<tr><td>Description:</td><td>Defines all the parameters for the ShownTickets object in the customer preferences of the customer interface.</td></tr>
<tr><td>Group:</td><td>Ticket</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Customer::Preferences</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'CustomerPreferencesGroups'}->{'ShownTickets'} = {
  'Active' => '1',
  'Column' => 'User Profile',
  'Data' => {
    '15' => '15',
    '20' => '20',
    '25' => '25',
    '30' => '30'
  },
  'DataSelected' => '25',
  'Key' => 'Tickets per page',
  'Label' => 'Number of displayed tickets',
  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
  'PrefKey' => 'UserShowTickets',
  'Prio' => '4000'
};
</pre></td></tr>
</table>

6.64.2. CustomerPreferencesGroups###RefreshTime

<table>
<tr><td>Description:</td><td>Defines all the parameters for the RefreshTime object in the customer preferences of the customer interface.</td></tr>
<tr><td>Group:</td><td>Ticket</td></tr>
<tr><td>SubGroup:</td><td>Frontend::Customer::Preferences</td></tr>
<tr><td>Valid:</td><td>1</td></tr>
<tr><td>Required:</td><td>0</td></tr>
<tr><td>Config-Setting:</td><td><pre>
 $Self->{'CustomerPreferencesGroups'}->{'RefreshTime'} = {
  'Active' => '1',
  'Column' => 'User Profile',
  'Data' => {
    '' => 'off',
</pre></td></tr>
</table>

	`'10' => '10 minutes',`
	`'15' => '15 minutes',`
	`'2' => ' 2 minutes',`
	`'5' => ' 5 minutes',`
	`'7' => ' 7 minutes'`
	`},`
	`'DataSelected' => '',`
	`'Key' => 'Refresh interval',`
	`'Label' => 'Ticket overview',`
	`'Module' => 'Kernel::Output::HTML::PreferencesGeneric',`
	`'PrefKey' => 'UserRefreshTime',`
	`'Prio' => '4000'`
	`};`

6.65. Frontend::Customer::Ticket::ViewNew

6.65.1. Ticket::Frontend::CustomerTicketMessage###NextScreenAfterNewTicket

Description:	Determines the next screen after new customer ticket in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'NextScreenAfterNewTicket'} =  'CustomerTicketOverview';`

6.65.2. Ticket::Frontend::CustomerTicketMessage###Priority

Description:	Allows customers to set the ticket priority in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'Priority'} = '1';`

6.65.3. Ticket::Frontend::CustomerTicketMessage###PriorityDefault

Description:	Defines the default priority of new customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'PriorityDefault'} =  '3 normal';`

6.65.4. Ticket::Frontend::CustomerTicketMessage###Queue

Description:	Allows customers to set the ticket queue in the customer interface. If this is set to 'No', QueueDefault should be configured.

Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'Queue'} =  '1';`

6.65.5. Ticket::Frontend::CustomerTicketMessage###QueueDefault

Description:	Defines the default queue for new customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'QueueDefault'} =  'Postmaster';`

6.65.6. Ticket::Frontend::CustomerTicketMessage###TicketType

Description:	Allows customers to set the ticket type in the customer interface. If this is set to 'No', TicketTypeDefault should be configured.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'TicketType'} = '1';`

6.65.7. Ticket::Frontend::CustomerTicketMessage###TicketTypeDefault

Description:	Defines the default ticket type for new customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'TicketTypeDefault'} =  'default';`

6.65.8. Ticket::Frontend::CustomerTicketMessage###Service

Description:	Allows customers to set the ticket service in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'Service'} = '1';`

6.65.9. Ticket::Frontend::CustomerTicketMessage###SLA

Description:	Allows customers to set the ticket SLA in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'SLA'} =  '1';`

6.65.10. Ticket::Frontend::CustomerTicketMessage###ServiceMandatory

Description:	Sets if service must be selected by the customer.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'ServiceMandatory'} =  '0';`

6.65.11. Ticket::Frontend::CustomerTicketMessage###SLAMandatory

Description:	Sets if SLA must be selected by the customer.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'SLAMandatory'} =  '0';`

6.65.12. Ticket::Frontend::CustomerTicketMessage###StateDefault

Description:	Defines the default state of new customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'StateDefault'} =  'new';`

6.65.13. Ticket::Frontend::CustomerTicketMessage###ArticleType

Description:	Defines the default type for article in the customer interface.

Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'ArticleType'} = 'webrequest';`

6.65.14. Ticket::Frontend::CustomerTicketMessage###SenderType

Description:	Sender type for new tickets from the customer inteface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'SenderType'} = 'customer';`

6.65.15. Ticket::Frontend::CustomerTicketMessage###HistoryType

Description:	Defines the default history type in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'HistoryType'} = 'WebRequestCustomer';`

6.65.16. Ticket::Frontend::CustomerTicketMessage###HistoryComment

Description:	Comment for new history entries in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'HistoryComment'} =  '';`

6.65.17. CustomerPanelSelectionType

Description:	Defines the receipent target of the tickets ("Queue" shows all queues, "SystemAddress" displays all system addresses) in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerPanelSelectionType'} =  'Queue';`

6.65.18. CustomerPanelSelectionString

Description:	Determines the strings that will be shown as receipent (To:) of the ticket in the customer interface. For Queue as CustomerPanelSelectionType, "<Queue>" shows the names of the queues, and for SystemAddress, "<Realname> <<Email>>" shows the name and email of the receipent.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerPanelSelectionString'} =  '<Queue>';`

6.65.19. CustomerPanelOwnSelection

Description:	Determines which queues will be valid for ticket's recepients in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	0
Required:	0

Config-Setting:	`$Self->{'CustomerPanelOwnSelection'} =  {`
	`  'Junk' => 'First Queue',`
	`  'Misc' => 'Second Queue'`
	`};`

6.65.20. CustomerPanel::NewTicketQueueSelectionModule

Description:	Module for To-selection in new ticket screen in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerPanel::NewTicketQueueSelectionModule'} =`
	`'Kernel::Output::HTML::CustomerNewTicketQueueSelectionGeneric';`

6.65.21. Ticket::Frontend::CustomerTicketMessage###DynamicField

Description:	Dynamic fields options shown in the ticket message screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required. NOTE. If you want to display these fields also in the ticket zoom of the customer interface, you have to enable them in CustomerTicketZoom###DynamicField.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew

Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'DynamicField'} =  {};`

6.66. Frontend::Customer::Ticket::ViewPrint

6.66.1. Ticket::Frontend::CustomerTicketPrint###DynamicField

Description:	Dynamic fields shown in the ticket print screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewPrint
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketPrint'}->{'DynamicField'} = {};`

6.67. Frontend::Customer::Ticket::ViewSearch

6.67.1. Ticket::CustomerTicketSearch::SearchLimit

Description:	Maximum number of tickets to be displayed in the result of a search in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomerTicketSearch::SearchLimit'} =  '5000';`

6.67.2. Ticket::CustomerTicketSearch::SearchPageShown

Description:	Number of tickets to be displayed in each page of a search result in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomerTicketSearch::SearchPageShown'} =  '40';`

6.67.3. Ticket::CustomerTicketSearch::SortBy::Default

Description:	Defines the default ticket attribute for ticket sorting in a ticket search of the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1

Required:	1
Config-Setting:	`$Self->{'Ticket::CustomerTicketSearch::SortBy::Default'} =  'Age';`

6.67.4. Ticket::CustomerTicketSearch::Order::Default

Description:	Defines the default ticket order of a search result in the customer interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomerTicketSearch::Order::Default'} =  'Down';`

6.67.5. Ticket::Frontend::CustomerTicketSearch###ExtendedSearchCondition

Description:	Allows extended search conditions in ticket search of the customer interface. With this feature you can search w. g. with this kind of conditions like "(key1&&key2)" or "(key1||key2)".
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'ExtendedSearchCondition'} =  '1';`

6.67.6. Ticket::Frontend::CustomerTicketSearch###SearchCSVData

Description:	Data used to export the search result in CSV format.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchCSVData'} =  [`
	`'TicketNumber',`
	`'Age',`
	`'Created',`
	`'Closed',`
	`'State',`
	`'Priority',`
	`'Lock',`
	`'CustomerID',`
	`'CustomerName',`
	`'From',`
	`'Subject'`
	`];`

6.67.7. Ticket::Frontend::CustomerTicketSearch###DynamicField

Description:	Dynamic fields shown in the ticket search screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled.

Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'DynamicField'} = {};`

6.67.8. Ticket::Frontend::CustomerTicketSearch###SearchOverviewDynamicField

Description:	Dynamic fields shown in the ticket search overview results screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchOverviewDynamicField'} =  {};`

6.67.9. Ticket::Frontend::CustomerTicketSearch###SearchCSVDynamicField

Description:	Dynamic Fields used to export the search result in CSV format.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchCSVDynamicField'} =  {};`

6.68. Frontend::Customer::Ticket::ViewZoom

6.68.1. Ticket::Frontend::CustomerTicketZoom###NextScreenAfterFollowUp

Description:	Determines the next screen after the follow up screen of a zoomed ticket in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'NextScreenAfterFollowUp'} =  'CustomerTicketOverview';`

6.68.2. Ticket::Frontend::CustomerTicketZoom###ArticleType

Description:	Defines the default type of the note in the ticket zoom screen of the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom

Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'ArticleType'} = 'webrequest';`

6.68.3. Ticket::Frontend::CustomerTicketZoom###SenderType

Description:	Defines the default sender type for tickets in the ticket zoom screen of the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'SenderType'} = 'customer';`

6.68.4. Ticket::Frontend::CustomerTicketZoom###HistoryType

Description:	Defines the history type for the ticket zoom action, which gets used for ticket history in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'HistoryType'} = 'FollowUp';`

6.68.5. Ticket::Frontend::CustomerTicketZoom###HistoryComment

Description:	Defines the history comment for the ticket zoom action, which gets used for ticket history in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'HistoryComment'} = '';`

6.68.6. Ticket::Frontend::CustomerTicketZoom###Priority

Description:	Allows customers to change the ticket priority in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'Priority'} =  '1';`

6.68.7. Ticket::Frontend::CustomerTicketZoom###PriorityDefault

Description:	Defines the default priority of follow up customer tickets in the ticket zoom screen in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'PriorityDefault'} =  '3 normal';`

6.68.8. Ticket::Frontend::CustomerTicketZoom###State

Description:	Allows choosing the next compose state for customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'State'} =  '1';`

6.68.9. Ticket::Frontend::CustomerTicketZoom###StateDefault

Description:	Defines the default next state for a ticket after customer follow up in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'StateDefault'} =  'open';`

6.68.10. Ticket::Frontend::CustomerTicketZoom###StateType

Description:	Defines the next possible states for customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'StateType'} =  [`
	`  'open',`
	`  'closed'`
	`];`

6.68.11. Ticket::Frontend::CustomerTicketZoom###AttributesView

Description:	Shows the activated ticket attributes in the customer interface (0 = Disabled and 1 = Enabled).
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'AttributesView'} =`
	`{`
	`'Owner' => '0',`
	`'Priority' => '1',`
	`'Queue' => '1',`
	`'Responsible' => '0',`
	`'SLA' => '0',`
	`'Service' => '0',`
	`'State' => '1',`
	`'Type' => '0'`
	`};`

6.68.12. Ticket::Frontend::CustomerTicketZoom###DynamicField

Description:	Dynamic fields shown in the ticket zoom screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'DynamicField'} =`
	`{};`

6.68.13. Ticket::Frontend::CustomerTicketZoom###FollowUpDynamicField

Description:	Dynamic fields options shown in the ticket reply section in the ticket zoom screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}-`
	`>{'FollowUpDynamicField'} =  {};`

6.69. Frontend::Customer::TicketOverview

6.69.1. Ticket::Frontend::CustomerTicketOverviewSortable

Description:	Controls if customers have the ability to sort their tickets.
Group:	Ticket
SubGroup:	Frontend::Customer::TicketOverview

Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketOverviewSortable'} =  '';`

6.69.2. Ticket::Frontend::CustomerTicketOverview###ColumnHeader

Description:	Shows either the last customer article's subject or the ticket title in the small format overview.
Group:	Ticket
SubGroup:	Frontend::Customer::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketOverview'}->{'ColumnHeader'} =  'TicketTitle';`

6.69.3. Ticket::Frontend::CustomerTicketOverview###Owner

Description:	Show the current owner in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::TicketOverview
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketOverview'}->{'Owner'} = '0';`

6.69.4. Ticket::Frontend::CustomerTicketOverview###Queue

Description:	Show the current queue in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::TicketOverview
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketOverview'}->{'Queue'} = '0';`

6.69.5. Ticket::Frontend::CustomerTicketOverview###DynamicField

Description:	Dynamic fields shown in the ticket overview screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Customer::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketOverview'}->{'DynamicField'} =  {};`

6.70. Frontend::Queue::Preferences

6.70.1. QueuePreferences###Comment2

Description:	Parameters of the example queue attribute Comment2.
Group:	Ticket
SubGroup:	Frontend::Queue::Preferences
Valid:	0
Required:	0

Config-Setting:	`$Self->{'QueuePreferences'}->{'Comment2'} =  {`
	`  'Block' => 'TextArea',`
	`  'Cols' => '50',`
	`  'Desc' => 'Define the queue comment 2.',`
	`  'Label' => 'Comment2',`
	`  'Module' => 'Kernel::Output::HTML::QueuePreferencesGeneric',`
	`  'PrefKey' => 'Comment2',`
	`  'Rows' => '5'`
	`};`

6.71. Frontend::SLA::Preferences

6.71.1. SLAPreferences###Comment2

Description:	Parameters of the example SLA attribute Comment2.
Group:	Ticket
SubGroup:	Frontend::SLA::Preferences
Valid:	0
Required:	0

Config-Setting:	`$Self->{'SLAPreferences'}->{'Comment2'} =  {`
	`  'Block' => 'TextArea',`
	`  'Cols' => '50',`
	`  'Desc' => 'Define the sla comment 2.',`
	`  'Label' => 'Comment2',`
	`  'Module' => 'Kernel::Output::HTML::SLAPreferencesGeneric',`
	`  'PrefKey' => 'Comment2',`
	`  'Rows' => '5'`
	`};`

6.72. Frontend::Service::Preferences

6.72.1. ServicePreferences###Comment2

Description:	Parameters of the example service attribute Comment2.
Group:	Ticket
SubGroup:	Frontend::Service::Preferences
Valid:	0
Required:	0

Config-Setting:	`$Self->{'ServicePreferences'}->{'Comment2'} =  {`
	`  'Block' => 'TextArea',`
	`  'Cols' => '50',`
	`  'Desc' => 'Define the service comment 2.',`
	`  'Label' => 'Comment2',`
	`  'Module' => 'Kernel::Output::HTML::ServicePreferencesGeneric',`
	`  'PrefKey' => 'Comment2',`
	`  'Rows' => '5'`

	`};`

Приложение C. Благодарности

OTRS is an open source project, and we would like to thank many people for their help and support. The following list is surely incomplete, and we apologize for that! Just drop us a note if you are not on this list.

Следующие люди внесли особый вклад в поект и являются его активными сторонниками:

- Robert Kehl, создавший Win32-инсталлятор для релиза 1.x. Больше тебе спасибо, Robert!

- Torsten Werner, который поддерживаетм Debian-installer для OTRS. Спасибо большое, Torsten, теперь OTRS также доступен для Debian-сообщества.

- Nils Jeppe (зеркало Гамбург, Германия), Bryan Fullerton (зеркало Торонто, Канада), Eberhard Mönkeberg (зеркало Геттинген, Германия), Timo Dreger (зеркало Дюссельдорф, Германия) и Netmonic (зеркало Вена, Австрия), которые поддерживают зеркальное копии нашего ftp-сервера. Большое спасибо, благодаря вашей помощи всегда есть возможность быстро скачать OTRS!

- Anja Schneider, которая помогает с переводом этого руководства и коррекции и пересмотр немецких текстов. Большое спасибо за вашу помощь и терпение, Anja!

- Через списки рассылки OTRS мы получаем множество идей, советов по усовершенствованию системы, обновления и исправления. Также здесь доступна большая поддержка для всех пользователей . Большое спасибо всем людям, которые участвуют в наших списках рассылки за вашу активную помощь!

Also we would like to extend 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 Ligtvoet
- Antti Kämäräinen
- Nicolas Goralski
- Gilberto Cezar de Almeida
- Jorge Becerra
- Eddie Urenda
- Stella Power
- Andreas Haase
- Reiner Keller
- Covert Jake
- Moshe Leibovitch
- Bjärn Jacke
- Remo Catelotti
- Alfons Obermeyer
- Michael Rech
- Danie Theron
- Richard Paradies
- Art Powell

Приложение D. GNU Free Documentation License

Version 1.1, March 2000

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

0. PREAMBLE

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

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

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

1. APPLICABILITY AND DEFINITIONS

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

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

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

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

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

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

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

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

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

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

3. COPYING IN QUANTITY

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

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

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

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

4. MODIFICATIONS

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

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version, as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

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

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

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

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

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

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

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

5. COMBINING DOCUMENTS

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

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

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

6. COLLECTIONS OF DOCUMENTS

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

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

7. AGGREGATION WITH INDEPENDENT WORKS

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

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

8. TRANSLATION

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

9. TERMINATION

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

10. FUTURE REVISIONS OF THIS LICENSE

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

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

. How to use this License for your documents

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

> Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation

> License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

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

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