

OTRS 3.1 - Administrator-Handbuch

OTRS 3.1 - Administrator-Handbuch

Copyright © 2003-2012 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 Oswald, Martha Elia Pascual, Thomas Raith, Carlos Fernando Rodríguez, Stefan Rother, Burchard Steinbild, Daniel Zamorano.

Dieses Werk ist geistiges Eigentum der OTRS AG.

Es darf als Ganzes oder in Auszügen kopiert werden, vorausgesetzt, dieser Copyright-Vermerk befindet sich auf jeder Kopie.

Der Quellcode dieses Dokuments kann unter source.otrs.org [<http://source.otrs.org/viewvc.cgi/doc-admin/>] abgerufen werden.

UNIX ist ein eingetragenes Warenzeichen von X/Open Company Limited. Linux ist ein eingetragenes Warenzeichen von Linus Torvalds.

MS-DOS, Windows, Windows 95, Windows 98, Windows NT, Windows 2000, Windows XP, Windows 2003 und Windows Vista sind eingetragene Warenzeichen der Microsoft Corporation. Andere Warenzeichen oder registrierte Warenzeichen: SUSE und YaST von SUSE Linux GmbH, Red Hat und Fedora von Red Hat Inc., Debian von Software in the Public Interest, Inc., Mandrake von MandrakeSoft, SA. MySQL und das MySQL Logo sind eingetragene Warenzeichen Oracle Corporation und/oder verbundener Unternehmen.

Alle Warennamen werden ohne Gewährleistung der freien Verwendbarkeit benutzt und sind möglicherweise eingetragene Warenzeichen.

Die Firma OTRS AG richtet sich im Wesentlichen nach den Schreibweisen der Hersteller. Andere hier genannte Produkte können Warenzeichen des jeweiligen Herstellers sein.

Inhaltsverzeichnis

Vorwort	xii
1. Trouble-Ticket-Systeme - die Grundlagen	1
Was ist ein Trouble-Ticket-System, und warum benötigen Sie eins?	1
Was ist ein Trouble-Ticket?	2
2. OTRS-Helpdesk	3
Grundlagen	3
Features	3
New features of OTRS 3.1	5
Die wichtigsten neuen Features von OTRS 3.0	9
Neue Features von OTRS 2.4	10
Hardware und Software-Anforderungen	12
Perl support	13
Web server support	13
Database support	13
Web browser support	13
Community	14
Kommerzieller Support und Dienstleistungen für OTRS	14
3. Installation	15
Der einfache Weg - Installation fertiger Pakete	15
Installation des RPM auf einem SuSE-Linux-Server	15
Installation von OTRS auf einem CentOS-System	17
Installation von OTRS auf einem Debian-System	17
Installation von OTRS auf einem Ubuntu-System	17
Installation von OTRS auf Microsoft Windows-Systemen	17
Manuelle Installation (Linux, Unix)	17
Vorbereiten der manuellen Installation	17
Installation der für OTRS benötigten Perl-Module	19
Konfiguration des Apache-Webservers	21
Einrichten der Datenbank	22
Einrichten der von OTRS benötigten cron-Jobs	29
Upgrade des OTRS Frameworks	33
Upgrade mit dem Windows-Installer	38
Upgrading Microsoft SQL Server Data Types	38
4. Erste Schritte	39
Agenten-Weboberfläche	39
Kunden-Weboberfläche	39
Öffentliche Weboberfläche	40
Die erste Anmeldung	41
Agenten-Weboberfläche - ein Überblick	41
Was verbirgt sich hinter dem Begriff Queue?	43
Benutzereinstellungen	44
5. Der Administrationsbereich von OTRS	47
Grundlagen	47
Benutzer, Gruppen und Rollen	47
Benutzer	47
Gruppen	48
Rollen	52
Kundenbenutzer und Kundengruppen	56
Kunden	56
Kundengruppen	57
Queues	59

Anreden, Signaturen, Anlagen und Antwortvorlagen	61
Anreden	61
Signaturen	62
Anlagen	64
Antwortvorlagen	66
Automatische Antworten	68
E-Mail-Adressen	71
Benachrichtigungen	72
S/MIME	74
PGP	75
Status	75
Die SysConfig (Verwaltung der Systemkonfiguration)	76
Einrichten von Mail-Konten	77
Einkommende Nachrichten filtern	78
Ausführen von automatisierten Jobs mit Hilfe des GenericAgents	81
Administrator-E-Mail	82
Sitzungsverwaltung	82
System Log	83
SQL-Abfragen mit Hilfe der SQL Box	84
Paketverwaltung	85
Web-Services	85
Dynamic Fields	86
6. Systemkonfiguration	87
Die Konfigurationsdateien von OTRS	87
Konfiguration des Systems mit Hilfe des grafischen Konfigurations-Frontends	87
7. E-Mails versenden/empfangen	89
E-Mails versenden	89
Via Sendmail (Standard)	89
Via SMTP server/smarthost	89
E-Mails empfangen	89
E-Mail-Konten, die über die OTRS-Oberfläche konfiguriert werden	89
Via Kommandozeilen-Programm und z. B. procmail (otrs.PostMaster.pl)	91
E-Mails via POP3 oder IMAP und fetchmail für otrs.PostMaster.pl empfangen..	92
Filterung/Verteilung über PostMaster-Module (für komplexere Verteilungsszenarien)	92
8. Zeitabhängige Funktionen in OTRS	94
Arbeitszeiten, Feiertage und Zeitzonen für das System festlegen	94
Arbeitszeiten	94
Feiertage mit festem Datum	94
TimeVacationDaysOneTime	94
Automatische Ticket-"Freigabe" (unlock)	95
9. Ticket-Verantwortlicher und Beobachten von Tickets	96
Ticket-Verantwortlicher	96
Tickets beobachten	97
10. Customizing the PDF output	99
11. Einbinden externer Backends	100
Kundenbenutzerdaten	100
Kundenbenutzer Backend	100
Datenbank (Standard)	100
LDAP	104
Verwenden mehrerer Kunden Backends	107
Backends für die Authentifizierung von Agenten und Kunden	110
Authentifizierungs-Backends für Agenten	111
Authentifizierungs-Backends für Kunden	113

Kunden-Selbstregistrierung anpassen	115
Anpassen der Weboberfläche	115
Kunden-Mapping	116
Anpassen der Kunden-Tabelle in der OTRS-Datenbank	118
12. Status	119
Vordefinierte Ticket-Status	119
Neu	119
Öffnen	119
Warten zur Erinnerung	119
Warten auf erfolglos schließen	119
Warten auf erfolgreich schließen	119
Zusammengefügt	119
Erfolgreich geschlossen	119
Erfolglos geschlossen	120
Anpassen von Ticket-Status	120
13. Anpassen der Ticket-Prioritäten	123
14. Erstellung eigener Themes	124
15. Übersetzung des OTRS-Frontends	125
16. PGP	126
17. S/MIME	130
18. Access Control Lists (ACLs)	135
Introduction	135
Examples	135
Reference	137
19. Statistikmodul	141
Bedienung des Moduls durch den Agent	141
Übersicht	141
Ansehen und Generieren von Statistiken	142
Bearbeiten / Neu	145
Importieren	150
Verwaltung des Statistik-Moduls durch den OTRS-Administrator	151
Rechte, Gruppen und Queues	151
Die SysConfig (Verwaltung der Systemkonfiguration)	152
Verwaltung des Statistik-Moduls durch den System-Administrator	152
Datenbanktabelle	152
Auflistung aller Dateien	152
Caching	153
otrs.GenerateStats.pl	153
Automatisiertes Generieren von Statistiken - Cron	153
Statische Statistiken	153
Wiederverwerten alter statischer Statistiken	154
Standardstatistiken	154
20. Generic Interface	155
Generic Interface Layers	155
Network Transport	155
Data Mapping	155
Controller	156
Operation (OTRS as a provider)	156
Invoker (OTRS as a requester)	156
Generic Interface Communication Flow	156
OTRS as Provider	156
OTRS as Requester	158
Web-Services	159
Web Service Graphical Interface	160

Web Service Overview	160
Web Service Add	160
Web Service Change	161
Web Service Command Line Interface	168
Web Service Configuration	169
Web Service Debugger	169
Web Service Configuration	170
Configuration Details	172
Connectors	177
Bundled Connectors	178
Examples:	188
21. OTRS Scheduler	193
Scheduler Graphical Interface	193
Scheduler Not Running Notification	193
Start Scheduler	194
Scheduler Command Line Interface	194
Unix / Linux	194
Windows	196
22. Dynamic Fields	199
Introduction	199
Configuration	199
Adding a Dynamic Field	200
Text Dynamic Field Configuration	202
Textarea Dynamic Field Configuration	202
Checkbox Dynamic Field Configuration	202
Dropdown Dynamic Field Configuration	203
Multiselect Dynamic Field Configuration	204
Date Dynamic Field Configuration	204
Date / Time Dynamic Field Configuration	205
Editing a Dynamic Field	206
Showing a Dynamic Field on a Screen	206
Setting a Default Value by a Ticket Event Module	
Set a Default Value by User Preferences	
Updating from OTRS 3.0	
23. Zusätzliche Applikationen	
FAQ	
24. Leistungsverbesserung	
OTRS	
TicketIndexModule	
TicketStorageModule	
Tickets archivieren	
Datenbank	
MySQL	
PostgreSQL	
Webserver	
Persistente Datenbankverbindungen	
Vorgeladene Module - startup.pl	
Perl Module bei Änderung neu laden	
Die richtige Strategie wählen	
mod_gzip/mod_deflate	
25. Datensicherung	
Datensicherung	
Wiederherstellung der Daten	
A. Weitere Quellen	

Homepage OTRS.org	
Mailinglisten	
Fehler berichten	
Kommerzieller Support	
B. Referenz der Konfigurationsoptionen	
Framework	
Core	
Core::LinkObject	
Core::Log	
Core::MIME-Viewer	
Core::MirrorDB	
Core::PDF	
Core::Package	
Core::PerformanceLog	
Core::ReferenceData	
Core::SOAP	
Core::Sendmail	
Core::Session	
Core::SpellChecker	
Core::Stats	
Core::Stats::Graph	
Core::Time	
Core::Time::Calendar1	
Core::Time::Calendar2	
Core::Time::Calendar3	
Core::Time::Calendar4	
Core::Time::Calendar5	
Core::Time::Calendar6	
Core::Time::Calendar7	
Core::Time::Calendar8	
Core::Time::Calendar9	
Core::Web	
Core::WebUserAgent	
Crypt::PGP	
Crypt::SMIME	
Frontend::Admin::AdminCustomerUser	
Frontend::Admin::ModuleRegistration	
Frontend::Agent	
Frontend::Agent::Dashboard	
Frontend::Agent::LinkObject	
Frontend::Agent::ModuleMetaHead	
Frontend::Agent::ModuleNotify	
Frontend::Agent::ModuleRegistration	
Frontend::Agent::NavBarModule	
Frontend::Agent::Preferences	
Frontend::Agent::SearchRouter	
Frontend::Agent::Stats	
Frontend::Customer	
Frontend::Customer::Auth	
Frontend::Customer::ModuleMetaHead	
Frontend::Customer::ModuleNotify	
Frontend::Customer::ModuleRegistration	
Frontend::Customer::Preferences	
Frontend::Public	

Frontend::Public::ModuleRegistration	
Ticket	
Core	
Core::FulltextSearch	
Core::LinkObject	
Core::PostMaster	
Core::Stats	
Core::Ticket	
Core::TicketACL	
Core::TicketBulkAction	
Core::TicketDynamicFieldDefault	
Core::TicketWatcher	
Frontend::Admin::ModuleRegistration	
Frontend::Agent	
Frontend::Agent::CustomerSearch	
Frontend::Agent::Dashboard	
Frontend::Agent::ModuleMetaHead	
Frontend::Agent::ModuleNotify	
Frontend::Agent::ModuleRegistration	
Frontend::Agent::Preferences	
Frontend::Agent::SearchRouter	
Frontend::Agent::Ticket::ArticleAttachmentModule	
Frontend::Agent::Ticket::ArticleComposeModule	
Frontend::Agent::Ticket::ArticleViewModule	
Frontend::Agent::Ticket::ArticleViewModulePre	
Frontend::Agent::Ticket::MenuModule	
Frontend::Agent::Ticket::MenuModulePre	
Frontend::Agent::Ticket::ViewBounce	
Frontend::Agent::Ticket::ViewBulk	
Frontend::Agent::Ticket::ViewClose	
Frontend::Agent::Ticket::ViewCompose	
Frontend::Agent::Ticket::ViewCustomer	
Frontend::Agent::Ticket::ViewEmailNew	
Frontend::Agent::Ticket::ViewEscalation	
Frontend::Agent::Ticket::ViewForward	
Frontend::Agent::Ticket::ViewFreeText	
Frontend::Agent::Ticket::ViewHistory	
Frontend::Agent::Ticket::ViewMailbox	
Frontend::Agent::Ticket::ViewMerge	
Frontend::Agent::Ticket::ViewMove	
Frontend::Agent::Ticket::ViewNote	
Frontend::Agent::Ticket::ViewOwner	
Frontend::Agent::Ticket::ViewPending	
Frontend::Agent::Ticket::ViewPhoneInbound	
Frontend::Agent::Ticket::ViewPhoneNew	
Frontend::Agent::Ticket::ViewPhoneOutbound	
Frontend::Agent::Ticket::ViewPrint	
Frontend::Agent::Ticket::ViewPriority	
Frontend::Agent::Ticket::ViewQueue	
Frontend::Agent::Ticket::ViewResponsible	
Frontend::Agent::Ticket::ViewSearch	
Frontend::Agent::Ticket::ViewStatus	
Frontend::Agent::Ticket::ViewZoom	
Frontend::Agent::TicketOverview	

Frontend::Agent::ToolBarModule	
Frontend::Customer	
Frontend::Customer::ModuleMetaHead	
Frontend::Customer::ModuleRegistration	
Frontend::Customer::Preferences	
Frontend::Customer::Ticket::ViewNew	
Frontend::Customer::Ticket::ViewPrint	
Frontend::Customer::Ticket::ViewSearch	
Frontend::Customer::Ticket::ViewZoom	
Frontend::Queue::Preferences	
Frontend::SLA::Preferences	
Frontend::Service::Preferences	
C. Danksagungen	
D. GNU Free Documentation License	
0. PREAMBLE	
1. APPLICABILITY AND DEFINITIONS	
2. VERBATIM COPYING	
3. COPYING IN QUANTITY	
4. MODIFICATIONS	
5. COMBINING DOCUMENTS	
6. COLLECTIONS OF DOCUMENTS	
7. AGGREGATION WITH INDEPENDENT WORKS	
8. Übersetzung	
9. TERMINATION	
10. FUTURE REVISIONS OF THIS LICENSE	
How to use this License for your documents	

Tabellenverzeichnis

3.1. Die folgenden Perl-Module werden von OTRS benötigt.	19
3.2. Die verschiedenen Skripte für die cron-Jobs von OTRS.	29
5.1. Standardmäßig vorhandene Gruppen in OTRS	49
5.2. Mit OTRS-Gruppen verknüpfte Rechte	51
5.3. Ereignisse für automatische Antworten	70
5.4. Funktion der verschiedenen X-OTRS-Header	78
21.1. List of Init Scripts And Supported Operating Systems	194
22.1. The following fields will be added into the system:	201
A.1. Mailinglisten	

Beispiele

5.1. Aussortierung von Spammails in eine bestimmte Queue	80
7.1. .fetchmailrc	92
7.2. Beispiel-Jobs für das Filtermodul Kernel::System::PostMaster::Filter::Match	92
7.3. Beispiel-Job für das Filtermodul Kernel::System::PostMaster::Filter::CMD	93
11.1. Konfiguration eines Datenbank-Kunden-Backends	100
11.2. Firmen-Tickets mit einem Datenbank-Backend	104
11.3. Konfiguration eines LDAP Kunden-Backends	104
11.4. Verwendung von Firmentickets mit einem LDAP-Backend	107
11.5. Gleichzeitige Einbindung mehrerer verschiedener Kunden Backends	107
11.6. Agentenauthentifizierung gegen ein Datenbank-Backend	111
11.7. Agentenauthentifizierung gegen ein LDAP Backend	111
11.8. Agentenauthentifizierung über HTTPBasic	113
11.9. Agentenauthentifizierung gegen ein Radius-Backend	113
11.10. Kundenauthentifizierung gegen ein Datenbank-Backend	113
11.11. Kundenauthentifizierung gegen ein LDAP Backend	114
11.12. Kundenauthentifizierung über HTTPBasic	115
11.13. Kundenauthentifizierung gegen ein Radius-Backend	115
18.1. ACL, die das Verschieben von Tickets in eine Queue nur bei einer Ticket-Priorität von 5 erlaubt.	135
18.2. ACL, die das Schließen von Tickets in der raw-Queue verbietet und den Schließen- Schalter ausblendet.	136
18.3. ACL removing always state closed successful.	137
18.4. ACL only showing Hardware services for tickets that are created in queues that start with "HW".	137
18.5. Reference showing all possible important ACL settings.	137
19.1. Festlegen einer Wertereihe bei einem Element	149
19.2. Festlegen einer Wertereihe bei zwei Elementen	149
21.1. Example To Start The OTRS Scheduler Form An Init.d Script	195
21.2. Example To Start The OTRS Scheduler	195
21.3. Example To Force Stop The OTRS Scheduler	196
21.4. Example To Register The OTRS Scheduler Into the Widows SCM	196
21.5. Example To Start The OTRS Scheduler	197
21.6. Example To Force Stop The OTRS Scheduler	197
22.1. Activate Field1 in New Phone Ticket Screen.	
22.2. Activate Field1 in New Phone Ticket Screen as mandatory.	
22.3. Activate several fields in New Phone Ticket Screen.	
22.4. Deactivate some fields in New Phone Ticket Screen.	
22.5. Activate Field1 in Ticket Zoom Screen.	
22.6. Activate Field1 in Ticket Overview Small Screens.	
22.7. Activate Field1 in TicketCreate event.	
22.8. Activate Field1 in the User preferences.	

Vorwort

Dieses Buch soll das Leben von OTRS-Administratoren und -Neulingen erleichtern.

Dieses Kapitel beschreibt die Installation, Konfiguration und Administration von OTRS. Das erste Drittel des Textes beschreibt die Hauptfunktionalität der Software, während der Rest als Referenz aller vorhandenen Konfigurationseinstellungen dient.

Da das Buch versucht, sich an den Bedürfnissen von OTRS Administratoren und OTRS Neulingen zu orientieren und da die Qualität der folgenden Kapitel so hoch wie möglich sein soll, sind wir auf Ihr Feedback angewiesen. Bitteteilten Sie uns mit, wenn Sie Abschnitte in diesem Buch vermissen, wenn Dinge für Sie unverständlich erklärt sind oder auch wenn Sie Rechtschreib-, Tipp- oder Grammatikfehler in diesem Buch entdecken. Jede Art von Rückmeldung ist ausdrücklich erwünscht und sollte durch einen Eintrag auf <http://bugs.otrs.org> an uns gerichtet werden, da sie so nicht verloren geht und direkt beim zuständigen Ansprechpartner landet. Wir bedanken uns schon jetzt für jede Art von Mithilfe!

Kapitel 1. Trouble-Ticket-Systeme - die Grundlagen

In diesem Abschnitt soll kurz die grundlegende Idee, die hinter Trouble-Tickets im Allgemeinen und Trouble-Ticket-Systemen im Speziellen steht, erläutert werden. An einem kleinen Beispiel wird gezeigt, wo die Vorteile dieser Systeme liegen.

Was ist ein Trouble-Ticket-System, und warum benötigen Sie eins?

Das folgende Beispiel soll verdeutlichen, was ein Trouble Ticket System ist und wie Sie damit in Ihrem Unternehmen Zeit und Geld einsparen können.

Nehmen wir an, dass Max Mustermann Fabrikant ist und Videorekorder produziert. Da die Programmierung der Videorekorder sehr unübersichtlich und kompliziert ist, wenden sich die Kunden von Herrn Mustermann gerne und häufig mit Supportanfragen per Mail an ihn. An manchen Tagen kann Herr Mustermann der Mailflut kaum Herr werden und so kommt es, dass seine Kunden sich einige Zeit gedulden müssen, bis die Antwort mit der rettenden Lösung bei ihnen eintrifft. Manchen Kunden dauert dies jedoch zu lange, eine weitere E-Mail mit dem gleichen Inhalt wird an Herrn Mustermann geschickt. Die E-Mails mit den Supportanfragen werden alle in eine INBOX weitergeleitet, wie sie von fast allen E-Mailprogrammen verwendet wird.

An manchen Tagen ist die Anfragewelle besonders groß und Herr Mustermann sieht sich außerstande, alle Mails noch in einem vertretbaren Zeitrahmen zu beantworten. Aus diesem Grund kommandiert er seine Entwickler Meier und Schulze zur Bearbeitung der Supportanfragen ab. Da von allen das gleiche System benutzt wird, greifen alle auf die gleiche INBOX und daher auch auf die gleichen Mails zu. Meier und Schulze haben jedoch keine Ahnung, dass manch ein Kunde in seiner Not gleich zwei E-Mails verfasst und an Herrn Mustermann geschickt hat. So kommt es vor, dass Meier die erste Mail mit einem anderen Ratschlag beantwortet als Schulze der sich im selben Moment der zweiten Nachricht des gleichen Kunden annimmt. Das Ergebnis ist, dass der Kunde unterschiedliche Antworten bekommt. Darüber hinaus hat Herr Mustermann keinen Einblick darüber, welcher Mitarbeiter wann was welchem Kunden gesagt hat, welche Probleme besonders häufig auftreten und wie groß sein gesamter Aufwand für den Kundensupport ist.

Von einem Kollegen erfährt Herr Mustermann schließlich, dass es Trouble-Ticket-Systeme gibt, die genau die Probleme lösen, die Herr Mustermann mit dem Support für seine Kunden hat. Herr Mustermann entscheidet sich für das offene Trouble-Ticket-Request System OTRS und installiert dieses System auf einem Rechner, der über einen Webserver sowohl für seine Mitarbeiter als auch über das Internet erreichbar ist. Von nun an werden die Hilferufe der Kunden nicht mehr länger an seine private INBOX, sondern direkt an den Mail-Account für OTRS weitergeleitet. OTRS hat eine Schnittstelle zur INBOX für die Supportanfragen, so dass alle ankommenden E-Mails automatisch ins Trouble Ticket System eingespeist werden. Unabhängig ob Herr Mustermann nun gerade anwesend ist oder nicht, generiert OTRS eine automatische Antwort und teilt dem Kunden mit, dass seine E-Mail angekommen ist und so schnell wie möglich bearbeitet wird. Dabei wird eine eindeutige Trouble Ticket Nummer vergeben. Der Kunde ist glücklich, dass sein Flehen schnell erhört wurde und wartet gespannt auf eine Antwort. Sowohl Herr Mustermann als auch die Entwickler Meier und Schulze können nun über einen beliebigen Internetbrowser und die Weboberfläche von OTRS auf die Supportanfragen zugreifen und diese

einzelnen beantworten. Da die Tickets bei Beantwortung gesperrt werden, wird keine Nachricht versehentlich zweimal erstellt.

Stellen wir uns vor, dass Herr Schmidt eine Anfrage ans System gestellt hat und Herr Meier diese kurz und knapp beantwortet. Herrn Schmidt reicht diese Antwort jedoch nicht aus und so antwortet er auf die Lösungsmail am folgenden Tag. Herr Meier ist jedoch gerade mit anderen Dingen beschäftigt, so dass sich Herr Mustermann der Sache annimmt. Über die History-Funktion von OTRS kann er jetzt auf alle vergangenen E-Mails von Herrn Schmidt und Herrn Meier zugreifen, deren Inhalt abfragen und eine ausführlichere Antwort versenden. Herr Schmidt erhält nun die Lösung für sein Problem, weiß aber nicht, dass diese von unterschiedlichen Personen stammt.

Natürlich ist dies nur ein sehr kleiner Einblick in die Funktionalitäten von Trouble Ticket Systemen. Da Herr Mustermann eine kleine Firma führt, erhält er vielleicht nur wenige E-Mails mit Supportanfragen pro Tag, die er vielleicht noch ganz überschaulich mit seiner normalen Mailsoftware handhaben kann und somit kein Trouble Ticket System braucht. Wenn aber der neue DVD-Rekorder in die Regale kommt, werden es vielleicht schon 500 oder in ein paar Jahren schon 10.000 Nachrichten pro Tag sein. Spätestens dann rechnet sich der Einsatz von Trouble-Ticket-Systemen wie OTRS.

Was ist ein Trouble-Ticket?

Ein Trouble-Ticket lässt sich im Wesentlichen mit einem Krankenblatt eines Krankenhauspatienten vergleichen. Bei der erstmaligen Einlieferung in das Krankenhaus wird das Krankenblatt im Zuge der Anamnese neu angelegt. Jeder Arzt trägt nun seine Diagnose sowie die verordnete Therapie und Medikation ein und dokumentiert deren Erfolg. Das Krankenblatt gibt nun einen schnellen Überblick, gewährleistet eine schnelle Einarbeitung und verhindert eine Mehrfachdosierung von Medikamenten. Ist die Krankheit besiegt und der Patient entlassen, wird das Krankenblatt archiviert.

Im OTRS werden Trouble Tickets, also die Krankenblätter aus dem obigen Beispiel, als normale E-Mails behandelt und gespeichert. Schickt z. B. ein Kunde eine Anfrage an das Trouble Ticket System, wird das Krankenblatt eingerichtet, ein neues Ticket wird geöffnet. Die Antwort eines Mitarbeiters auf die Anfrage kann als Eintrag eines Arztes gesehen werden, eine erneute Antwort bzw. Anfrage des Kunden auf das selbe Ticket als Veränderung oder Erweiterung des Krankheitsbildes. Ein Ticket gilt als erledigt bzw. geschlossen, wenn eine Antwort auf die Anfrage an den Kunden zurückgesendet wurde oder das Ticket über das System als geschlossen markiert wird. Antwortet ein Kunde auf ein bereits geschlossenes Ticket, so wird es erneut geöffnet und die neuen Informationen ergänzt. Um die Konsistenz der Daten sicherzustellen, werden alle Tickets mit all ihren spezifischen Informationen archiviert und verbleiben im System. Durch die Speicherung der Tickets als ganznormale E-Mails ist es möglich, dass diese auch E-Mail-Anhänge enthalten können. Zusätzlich zu den normalen Informationen einer E-Mail lassen sich beliebige Notizen zu jedem Ticket hinzufügen. Die Tickets selbst werden auf der Festplatte bzw. in einer Datenbank archiviert, ebenso zusätzliche Meta-Informationen des Tickets wie Notizen, an der Beantwortung des Tickets beteiligte Mitarbeiter, Zeit und Datum der Bearbeitung, Bearbeitungsdauer usw. Eine Sortierung oder eine Suche über den Datenbestand wird mit Hilfe aller vorhandenen Informationen zu den Tickets realisiert.

Kapitel 2. OTRS-Helpdesk

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

Grundlagen

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

OTRS is separated into several components. The basic component is the OTRS framework that contains all central functions for the application and the ticket system. Via the web interface of the OTRS framework, it is possible to install additional applications such as ITSM modules, integrations with Network Monitoring solutions, a knowledge base (FAQ), et cetera.

Features

OTRS bietet viele verschiedene Features. Die folgende Aufzählung gibt einen Überblick über die wichtigsten Eigenschaften und Fähigkeiten des zentralen OTRS-Frameworks.

Die Features von OTRS

- Web-Interface:
 - Easy and initial handling with any modern web browser, even with mobile phones or other mobile computers.
 - Eine Web-Oberfläche zur Administration des Systems ist vorhanden.
 - Ein Webinterface für die Mitarbeiter (Agenten) zur Bearbeitung von Kundenanfragen ist verfügbar.
 - Eine Weboberfläche für Kunden, über die Nachrichten an zuständige Agenten geschickt werden können und der Status eigener Tickets abgerufen werden kann, ist vorhanden.
 - Unterstützung für verschiedene Oberflächen-Layouts (Themes).
 - Unterstützung vieler Sprachen.
 - Eigene Anpassungen der Ausgabe-Vorlagen sind möglich (DTL).
 - Mehrfach-Anhänge von Emails sind über die Weboberfläche möglich.
- E-Mail-Schnittstelle:
 - Unterstützung für E-Mail-Anhänge (MIME)
 - Automatische Umwandlung von HTML- in reine Text-Nachrichten (höhere Sicherheit vor schädlichen Inhalten und schneller durchsuchbar).
 - Filterung von E-Mails über eigene X-Header-Einträge oder Mailadressen, z. B. für die Aussortierung von Spam.

- PGP-Support, Erstellung und Import eigener Zertifikate, verschlüsselter und signierter Mails, Anzeige von verschlüsselten und signierten Nachrichten.
- Unterstützung für die Verschlüsselung und Anzeige von SMIME-Nachrichten.
- Automatisierte Antworten (auto responder) für die Benachrichtigung von Kunden, abhängig von der Queue konfigurierbar.
- E-Mail-Benachrichtigungen für Agenten über neue Tickets, Follow-ups oder freigegebene Tickets.
- Follow-Ups anhand von Reference- oder In-Reply-To-Headern, automatisierte Zuordnung von Follow-Ups über den Ticket-Bezeichner im Betreff, Mailbody oder in Text-Anhängen.
- Tickets:
 - Erweiterte Queue-Ansicht, Übersicht über alle Anfragen innerhalb einer Queue.
 - Sperren von Tickets.
 - Erstellung eigener Antwortvorlagen.
 - Erstellung eigener auto responder, abhängig von der Queue.
 - Ticket-History, Übersicht über die komplette Entwicklung eines Tickets, Änderungen der Ticketstatus, Übersicht über die verschiedenen Aktionen für ein Ticket usw.
 - Druckansicht für Tickets, Export als PDF möglich.
 - Hinzufügen eigener (interner oder externer) Notizen zu einem Ticket (eigener Text und Dateianhänge).
 - Möglichkeit zum Zoomen von Tickets.
 - Definition von ACLs (access control lists) für Tickets.
 - Tickets können an andere E-Mail-Adressen weiter- oder umgeleitet werden (forwarding, bouncing).
 - Verschieben von Tickets zwischen verschiedenen Queues.
 - Festlegen der Priorität für Tickets.
 - Erfassung der Bearbeitungsdauer für Tickets.
 - Anstehende Aufgaben für ein Ticket festlegen (pending features).
 - Massenoperationen auf Tickets sind möglich (bulk features).
 - Automatische und zeitgesteuerte Aktionen können mit Hilfe eines sog. "GenericAgent" auf Tickets ausgeführt werden.
 - Volltextsuche über den gesamten Ticketbestand und Export der Suchergebnisse als PDF.
- System:

- OTRS läuft unter vielen Betriebssystemen (Linux, Solaris, AIX, FreeBSD, OpenBSD, Mac OS 10.x, Microsoft Windows).
- Unterstützung von ASP (active service providing).
- Verknüpfung von Objekten wie z. B. Tickets, FAQ-Einträgen o.ä. innerhalb des Systems.
- Einbindung externer Datenquellen für die Kundendaten, z. B. über AD, eDirectory oder OpenLDAP).
- Festlegen einer eigenen Kennzeichnung für Tickets, z. B. Call#, Ticket#, Request# o.ä.
- Festlegen einer eigenen Nummerierung für Tickets.
- Support of several database systems for the central OTRS back-end, e.g. MySQL, PostgreSQL, Oracle, MSSQL).
- Framework für die Erstellung von Statistiken über die Web-Oberfläche, Export und Import von Reports und Statistikmodulen, Export von Statistiken als PDF ist möglich.
- Unicode-Unterstützung für Front- und Back-End.
- Die Authentifikation für Agenten oder Kunden kann unabhängig voneinander über eine Datenbank, LDAP, HTTPAuth oder Radius realisiert werden.
- Unterstützung von Benutzer-Accounts, Benutzergruppen und Rollen.
- Unterstützung verschiedener Zugriffsrechte, z. B. auf Queues oder Systembereiche.
- Die Erstellung von Standardantworten ist möglich.
- Unter-Queues werden unterstützt.
- Anreden und Signaturen können abhängig von der Queue definiert werden.
- E-Mail-Benachrichtigungen für Administratoren.
- Bekanntgabe von Informationen zu Updates über die Weboberfläche oder via E-Mail.
- Festlegen von Ablauffristen für problematische Tickets.
- Benutzerabhängige Unterstützung für verschiedene Zeitzonen.
- Einfache Einbindung eigener Addons und Module mit Hilfe der OTRS API.
- Einfache Erstellung eigener Front-Ends, z. B. X11, Console usw.

New features of OTRS 3.1

1 GENERIC INTERFACE - A Web Service Framework

- GI is a flexible framework to allow web service interconnections of OTRS with third party applications.
- OTRS can act in both ways - as a provider (server, requested from remote) or requester (client, requesting remotely).

- Simple web service connections can be created without programming by configuring the Generic Interface.
- Complex scenarios can be realized by plugging in custom OTRS extensions that add perl code to the GI infrastructure on different architectural layers.
- *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 agents 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:

- A new web service in the admin GUI
- The basic meta data (Transport type (SOAP), URL etc.) and
- Existing operations (part of a connector) and specify for each operation how the data must be mapped (inbound and outbound)

A Generic Interface Debugger will help the OTRS administrator to check how requests are coming in and how they are handled through the different layers.

1.1 Current Features

- Network transports: SOAP/HTTP. Others like REST and JSON are scheduled to be added in the future depending on customers demand.
- Configurable data mapping Graphical User Interface for key/value transformations with respect to incoming and outgoing data.
- Graphical debugger to check the configuration and flow of information of configured web services.
- A ticket connector allowing the use of OTRS as a web service for ticket handling.

1.2 Future Features

- Additional network transports (REST, JSON).
- The GI will replace the iPhoneHandle as the backend for mobile apps.
- 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 DYNAMIC FIELDS

The DynamicFields Feature replaces the existing ticket and article FreeText and FreeTime fields with a dynamic structure that will also allow to create custom forms in OTRS.

- An unlimited amount of fields can be configured using an own graphical user interface for administration.

- The fields can have different types that can be used for both, tickets and articles. Available by default are:
 - Text
 - Multiline text
 - Checkbox
 - Dropdown
 - Multi-select
 - Date
 - Date and time
- New custom field types (e.g. custom field type dropdown with an external data source) can be added with small effort as the fields are created in a modular, pluggable way.
- 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.
- A database update script will transform historic FreeText fields and related configuration settings into the new structure.

3 TICKET MANAGEMENT IMPROVEMENTS

3.1 Ticket creation improved

- Multiple email addresses can now be specified as 'To:', 'CC:' or 'BCC:' when creating a new phone or email ticket.

3.2 Inbound phone call support

- Inbound phone calls can now be registered within an existing tickets (until now, only outbound calls were registered).

3.3 Ticket overview preview improved

- It is now possible to exclude articles of certain sender types (e.g. articles from internal agents) in the SysConfig from being displayed in the overview preview mode.
- A certain article type can be configured which will display articles of that type as expanded by default when the view is accessed.

3.4 Ticket move improved

- The screen shown after moving a ticket is now configurable. Options are the ticket zoom view (LastScreenView) or the ticket list (LastScreenOverview).

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.

- An additional bulk action allows configuring the ticket type for selected tickets.

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.

4 PROCESS AUTOMATION

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.

4.2 Notification mechanism improved

- A new generic agent notification module allows the OTRS administrator to define messages that will be shown in the agent web front-end when agents log into the system.

4.3 Time calculation improved

- All kind of times are from now on 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.

4.4 GenericAgent improved

- The GenericAgent can now filter for tickets change time.
- In addition, the GenericAgent can set the ticket responsible for matched tickets.

5 USER INTERFACE, RICH TEXT EDITOR, CHARSET

5.1 User interface performance improved

- The speed for rendering and article display was improved, thanks to Stelios Gikas <stelios.gikas@noris.net>!

5.2 Rich Text Editor Update

- IOS5 support added.
- Block quotes can be left with the enter key.
- Update from CKEditor 3.4 to CKEditor 3.6, so improvements refer to the releases of CKEditor 3.5 [http://ckeditor.com/blog/CKEditor_3.5_released] and CKEditor 3.6 [http://ckeditor.com/blog/CKEditor_3.6_released].
- IE9 support improved.
- Resizable dialogs.

5.3 Unicode Support - Non-UTF-8 Internal Encodings Dropped

- UTF-8 is now the only allowed internal charset of OTRS.

- All language files are now formatted in UTF-8, which simplifies their handling and future improvements of the translation mechanism.

6 DATABASE DRIVER SUPPORT

6.1 PostgreSQL DRIVER compatibility improved

- PostgreSQL 9.1 support added.
- A new legacy driver is now available for PostgreSQL 8.1 or earlier versions.

6.2 MS SQL DRIVER compatibility improved

- The MS SQL driver now stores binary data in VARBINARY rather than deprecated type TEXT as well as NVARCHAR to store text strings rather than VARCHAR (for improved Unicode support).

7 MAIL INTEGRATION

7.1 Mail handling improved

- When connecting to IMAP mail accounts, it is now possible to handle emails from a specific email folder, other than the INBOX folder.
- OTRS can now also connect to IMAP servers using Transport Layer Security (TLS), useful for modern restricted environments.

Die wichtigsten neuen Features von OTRS 3.0

Kontext

- Benutzerzentriertes Redesign der graphischen Benutzerschnittstelle, welches einen Übergang zu dynamischen Technologien wie AJAX, XHTML und CSS beinhaltet.

Hervorhebung für "neue Tickets" und "neue Artikel"

- Dieses neue Feature wurde für Tickets und Artikel implementiert. Es erlaubt einem Agenten, auf einen Blick zu erkennen, ob zu bestimmten Tickets oder auch innerhalb eines Artikels neue, ungelesene Artikel gibt. Sie profitieren durch erhöhte Transparenz und kürzere Reaktionszeiten.

Optimierte Volltextsuche

- Das neue Such-Feature erlaubt es, die vorhandenen Informationen flexibel zu durchforsten. Die Optionen reichen von einfachem Suchwort bis hin zu komplexen, multi-Wort-fähigen booleschen Suchoperationen, in denen man verschiedene Operatoren verwenden kann. Somit stehen ihnen gemäß Ihrer Anforderungen anpassbare Suchmöglichkeiten zur Verfügung.

Neue Ticket-Zoom-Ansicht

- Die neue, AJAX-basierte Ticket-Zoom-Ansicht erlaubt die Darstellung komplexer und verknüpfter Informationen in Echtzeit unter Beibehaltung der aktuellen Arbeitsumgebung

des Agenten. Letzterer profitiert von verbesserter Orientierung und daraus folgend höherer Arbeitseffizienz.

Globale Ticketübersichten

- Die seit OTRS 2.4 wohlbekannten Ticketübersichten wurden für verbesserte Interaktivität optimiert. Je nach Anwendungsszenario können Ihre Agenten die Ticketübersicht leicht wechseln. Klein, Mittel und Groß stehen zur Verfügung und stellen jeweils eine unterschiedliche Detailtiefe bereit.

Accessibility

- Das Redesign berücksichtigt die anerkannten Zugänglichkeits-Standards WCAG und WAI-ARIA. Dadurch können behinderte Nutzer besser mit OTRS arbeiten. Die Bestimmungen des Abschnitt 508 des US Rehabilitation Act werden erfüllt.

Neues Kunden-Interface

- Das Kunden-Web-Interface kann in das Intranet Ihrer Organisation integriert werden und ist eng mit dem neuen OTRS-System verzahnt.

Archivierungs-Feature

- Mit OTRS 3.0 können Sie Tickets archivieren. Dadurch verkürzen sich Zeiten für Suchabfragen und deren Ergebnisauswertung.

Neue Features von OTRS 2.4

Lizenzwechsel auf die AGPL Version 3

- Warum AGPL statt GPL? AGPL und GPL sind bis auf einen einzigen Punkt identisch. Wird Software im Bereich SaaS eingesetzt, greift bei der AGPL das Copyleft, bei der GPL nicht. Wir möchten, dass auch im wachsenden SaaS-Umfeld Weiterentwicklungen an OTRS in die Community zurückfließen.

Warum v3 statt v2? Die GPL v2 ist in die Jahre gekommen und hat insbesondere außerhalb der USA diverse rechtliche Unsicherheiten. Die GPLv3 bzw. AGPLv3 ist unter den Open Source Lizenzen diejenige Copyleft-Lizenz, die sowohl den Copyright-Inhabern als auch dem Anwender den besten Schutz und die größte Rechtssicherheit bietet.

Neues Management-Dashboard

- Der Bedarf an einer systemweiten, personalisierten und aktuellen Präsentation wichtiger Informationen hat zur Entwicklung des Management-Dashboards geführt. Dafür können Plugins erstellt werden, die Inhalte von Erweiterungsmodulen zusätzlich zu den standardmäßig verfügbaren Informationen darstellen. Standard-Plugins sind:
- Ticket-Volumen (neu & offen) der letzten 24h, 48h und 72h
- Kalender mit anstehenden Ereignissen (Eskalationen, automatische Ticketentsperrungen usw.)
- Systemweite Übersicht der Ticketverteilung in den Queues
- Zeit für die erste Reaktion / Lösungszeit von Queues

- RSS-Integration

Weitere Standard-Reports

- Die zusätzlichen Standardreports von OTRS 2.4 sind:
- Erstellte Tickets
- Geschlossene Tickets
- SLA-Analyse
- Benötigte Bearbeitungszeit pro Kunde / pro Queue
- Analyse der Lösungszeit pro Kunde / pro Queue
- Analyse der Antwortzeit pro Kunde / pro Queue

Neues Master/Slave Ticket-Feature

- Die Master Slave Funktion erlaubt Ihnen, Tickets zum gleichen Thema zu markieren, um sie über das Master-Ticket gebündelt abzuarbeiten. Die Slave-Tickets werden automatisch gelöst und Ihre Kunden mit der im Master-Ticket hinterlegten Antwort benachrichtigt, wenn das Master-Ticket auf gelöst gesetzt wird.

Alle Slave-Tickets, 'erben' die folgenden Aktionen ihres Masters:

- Status ändern
- Email-Antworten
- Freitextfelder verändern
- Anmerkungen
- Wartezeit ändern
- Priorität ändern
- Besitzer ändern
- Verantwortlichen ändern

HTML E-Mail Unterstützung

- Erstellen und formatieren Sie E-Mails, Notizen & Benachrichtigungen nun auch im Rich-Text-Format (HTML-Format) oder binden Sie Bilder über der WYSIWYG-Editor ein.

Neue Out-Of-Office-Funktion

- Definieren Sie über Ihre persönlichen Einstellungen den Zeitraum Ihrer Abwesenheit. Aktivierung hat folgende Auswirkungen:

In Auswahllisten, die der Zuweisung eines Besitzers oder Verantwortlichen dienen, wird die Abwesenheitszeit und -dauer hinter Ihrem Namen dargestellt.

Bei FollowUps, wird das Ticket vollautomatisch "entsperrt" und das Bearbeitungs-Team der Queue erhält eine Benachrichtigung.

Neue Ticket-Übersichten und globale Sammelaktion

- Flexibel zur Laufzeit gestaltbare Ticket-Übersichten erleichtern Ihnen die tägliche die Arbeit. Ob tabellarische Übersicht oder Ticket-Vorschau, ein Small / Medium / Large View Icon erlaubt es Ihnen, sich je nach Einsatzszenario Ihre Übersichten, z. B. den Queue View, Status View, Escalation View oder Suchergebnisse individuell einzurichten.

Die systemweit in allen Ticket-Übersichten integrierte Sammelaktion erlaubt Ihnen die gleichzeitige Bearbeitung mehrerer Tickets.

Postmaster-Filter erkennen Follow-Ups zu intern weitergeleiteten Nachrichten

- Derzeit werden E-Mail-Antworten auf weitergeleitete Artikel in OTRS als "email-extern" aufgenommen. Das Problem dabei ist, dass diese Antworten vom Kunden im Web-Interface eingesehen werden können. Zwar wäre es möglich, E-Mails generell als "email-intern" zu klassifizieren, doch dann können Kunden nicht mehr angemessen bedient werden, weil sie ihre Tickets im Web-Interface nicht mehr korrekt nachverfolgen könnten. Mit diesem neuen Feature können E-Mail-Antworten zurückverfolgt werden, und der Artikeltyp "email-extern" oder "email-intern" wird je nach dem Typ des ursprünglichen Artikels automatisch gesetzt.

Konfigurierbare Event-basierte Benachrichtigungen

- Um eine höhere Flexibilität für E-Mail-Benachrichtigungen zu erreichen, wurde ein komplett neuer Benachrichtigungs-Mechanismus implementiert. Dieser ermöglicht es, event-basierte Benachrichtigungen an Agenten, Kunden oder dedizierte Email-Adressen zu versenden.

Somit ist es einfach möglich, nur beim Schließen eines Tickets eine E-Mail-Benachrichtigung an den Kunden eines Tickets zu senden. Oder z. B. beim Erstellen von VIP Tickets für einen bestimmten Kundenkreis mit sehr hoher Priorität, eine bestimmte E-Mail-Adresse (z. B. ein Bereitschaftshandy) zu benachrichtigen. Die dabei jeweils zu berücksichtigenden Events (z. B. Erstellung neuer Tickets, Ticketstatus-Update, Queue-Update, Erstellung neuer Artikel, usw.) und Inhalt der Benachrichtigungen (inkl. OTRS-Platzhalter) sind frei über ein Web-Interface konfigurierbar.

Leserechte und Benachrichtigungen auf Tickets der Watchliste

- Mit der "Read-Only"-Feature haben Sie Leserechte auf ein Ticket in Ihrer Watchliste, auch wenn das Ticket in eine Queue verschoben wurde, in der Sie keine Leserechte haben. Über Ihre persönlichen Einstellungen haben Sie die Möglichkeit zu definieren, ob Sie bei beobachteten Tickets, wie ein Besitzer oder Verantwortlicher über Änderungen per E-Mail benachrichtigt werden soll.

Secure SMTP-Unterstützung

- Beim Empfang von E-Mails unterstützt OTRS bislang POP3, POP3s, IMAP und IMAPs. Beim Versand neben lokalen Mail-Transfer-Agents (z. B. Sendmail, Postfix, Exim) und SMTP nun auch SMTPS (Secure-SMTP).

Hardware und Software-Anforderungen

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 deploy 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, on the OTRS machine, you should install perl and/or install some additional perl modules. The web server and Perl have to be installed on the same machine as OTRS. The database back-end can be installed locally or on another host.

For the web server we recommend using the Apache HTTP Server, because its module `mod_perl` improves greatly 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.

Für Perl gilt mindestens die Version 5.8.8 zu verwenden. Es werden einige Zusatzmodule benötigt, die Sie entweder direkt über die Shell von Perl und CPAN oder mit Hilfe des Paketmanagers (yast, apt-get) Ihres Betriebssystems einspielen müssen.

Software-Anforderungen

Perl support

- Perl 5.8.8 oder höher

Web server support

- Apache2 und `mod_perl2` (`mod_perl2` wird wegen der hohen Geschwindigkeit empfohlen)
- Webserver mit CGI support (CGI nicht empfohlen)
- Microsoft Internet Information Server (IIS) 6 or higher

Database support

- MySQL 4.1 oder höher
- PostgreSQL 7.0 or higher (8.2 or higher recommended)
- Oracle 10g oder höher
- Microsoft SQL Server 2005 or higher

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

Wenn Sie ein bereits vorgefertigtes OTRS-Paket für Ihr Betriebssystem zur Installation verwenden (rpm, Windows-Installer), sollten die benötigten Perl-Module automatisch installiert werden.

Web browser support

For the Agent interface of OTRS, you'll be OK if you use a modern browser with JavaScript support enabled. We support the following browsers:

- Internet Explorer 8.0 or higher
- Mozilla Firefox 3.6 or higher
- Google Chrome
- Opera 10 or higher
- Safari 4 or higher

We recommend to always use 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.

For the OTRS Customer Interface, in addition to the browsers listed above, you can also use Internet Explorer versions 6 or 7, and we do not require JavaScript either.

Community

OTRS has a large user community. Users and developers discuss about OTRS and interchange 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.

The homepage of the OTRS community is: <http://www.otrs.com/open-source/>.

Kommerzieller Support und Dienstleistungen für OTRS

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

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

You can find more detailed information about OTRS Group on <http://www.otrs.com> and you can contact us via email on [sales at otrs.com](mailto:sales@otrs.com) [<mailto:sales@otrs.com>].

Kapitel 3. Installation

Dieser Abschnitt beschreibt die Installation und die grundlegende Einrichtung des zentralen OTRS Frameworks. Dabei wird die Installation von OTRS aus dem Quellcode erläutert, wie auch über Binärpakete wie RPM oder ausführbare Dateien für Windows.

Dieses Kapitel beschreibt die Einrichtung des Web- und Datenbankservers, die Schnittstelle zwischen OTRS und der Datenbank, das Einspielen einzelner Perl-Module, das Setzen der richtigen Berechtigungen, die Einrichtung der OTRS-eigenen cron-Jobs sowie grundlegende Einstellungen in den OTRS-Konfigurationsdateien.

Folgen Sie den Schritten in diesem Kapitel, um OTRS auf Ihrem Server zu installieren. Dann können Sie sich über die Weboberfläche anmelden und das System administrieren.

Der einfache Weg - Installation fertiger Pakete

Der einfachste und komfortableste Weg ein lauffähiges OTRS zu installieren ist sicherlich, auf bereits vorgefertigte Pakete zurück zu greifen. Viele bereits vorgefertigte Installations-Pakete sind im Download-Bereich unter <http://www.otrs.org> zu finden. Da der Aufwand viel zu groß wäre, die Installation aller dort aufgeführten Pakete in dieser Dokumentation anzuführen, soll im Folgenden nur näher auf die Installation von OTRS unter SUSE Linux, Debian und Microsoft Windows eingegangen werden. Sehen Sie unter der o.g. URL nach, ob auch für Ihr Betriebssystem ein fertiges Installations-Paket vorhanden ist und greifen Sie nur auf die manuelle Installation zurück, wenn Sie keine andere Möglichkeit haben.

Installation des RPM auf einem SuSE-Linux-Server

This section demonstrates the installation of a pre-built RPM package on a SUSE Linux distro. We have tested against all recent SLES and openSUSE versions. Before you start the installation, please have a look at <http://www.otrs.org/downloads> [<http://www.otrs.org/downloads>] and check if a newer OTRS RPM package is available. Always use the latest RPM package.

Installieren Sie OTRS mittels yast (yast2) oder der Kommandozeile und **rpm**, je nach Vorliebe. Beachten Sie jedoch, dass OTRS einige Perl-Module benötigt, die nicht standardmäßig in einer typischen SUSE-Installation enthalten sind. yast sollte die bessere Wahl sein, da es alle Abhängigkeiten automatisch beachtet und auflösen kann.

Sollten Sie den Weg über die Kommandozeile mit **rpm** bevorzugen, so müssen Sie die Perl-Module manuell vor Beginn der Installation von OTRS installieren. Angenommen Sie haben die Datei `otrs.rpm` im Verzeichnis `/tmp` gespeichert, dann geben Sie zur Installation von OTRS folgenden Befehl ein.

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

Next steps:

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

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

[install the OTRS database]
Use a web browser and open this link:
http://localhost/otrs/installer.pl

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

Have fun!

Your OTRS Team
http://otrs.org/

linux:~ #
```

Skript: Kommando für die Installation von OTRS.

Nach der Installation des rpm's ist es notwendig, SuSEconfig zu starten. Geben Sie hierzu Folgendes ein.

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

Skript: Ausführung des Kommandos SuSEconfig.

Die Installation von OTRS ist abgeschlossen. Starten Sie nun Ihren Webserver neu, um die Änderungen in der Konfiguration zu übernehmen, wie im Skript unten gezeigt.

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

Skript: Neustart des Webservers.

Der nächste Schritt ist das Aufsetzen der Datenbank wie in dem Abschnitt über die Datenbankkonfiguration beschrieben.

Installation von OTRS auf einem CentOS-System

On the OTRS Wiki you can find detailed instructions for setting up OTRS on a CentOS system. Please note that these instructions will also apply to Red Hat Linux systems since they use the same source: http://wiki.otrs.org/index.php?title=Installation_of_OTRS_3.0b1_on_CentOS_5.5 [http://wiki.otrs.org/index.php?title=Installation_of_OTRS_3.0b1_on_CentOS_5.5].

Installation von OTRS auf einem Debian-System

Im OTRS-Wiki finden Sie detaillierte Anweisungen für die Installation unter Debian: http://wiki.otrs.org/index.php?title=Installation_on_Debian_5.04_lenny [http://wiki.otrs.org/index.php?title=Installation_on_Debian_5.04_lenny].

Installation von OTRS auf einem Ubuntu-System

Im OTRS-Wiki finden Sie detaillierte Anweisungen für die Installation unter Ubuntu: [http://wiki.otrs.org/index.php?title=Installation_on_Ubuntu_Lucid_Lynx_\(10.4\)](http://wiki.otrs.org/index.php?title=Installation_on_Ubuntu_Lucid_Lynx_(10.4)) [[http://wiki.otrs.org/index.php?title=Installation_on_Ubuntu_Lucid_Lynx_\(10.4\)](http://wiki.otrs.org/index.php?title=Installation_on_Ubuntu_Lucid_Lynx_(10.4))].

Installation von OTRS auf Microsoft Windows-Systemen

Die Installation von OTRS unter Microsoft Windows ist denkbar einfach. Laden Sie den auf <http://www.otrs.org/downloads/> bereit gestellten Installer herunter und speichern Sie die Datei. Anschließend führen Sie den Installer einfach aus und folgen den einzelnen Installationsschritten. Danach können Sie sich als OTRS-Administrator am System anmelden und dieses weiter konfigurieren. Um sich als Administrator anzumelden, verwenden Sie den Benutzernamen 'root@localhost' und das voreingestellte Passwort 'root'.

Warnung

Bitte ändern Sie das Passwort für den Benutzer 'root@localhost' umgehend.

Wichtig

Der Windows-Installer für OTRS beinhaltet bereits alle Komponenten, die für den Betrieb von OTRS benötigt werden. D.h., es wird zusätzlich zum eigentlichen OTRS der apache2 Webserver, MySQL, Perl mit den für OTRS benötigten Modulen und cron für Windows installiert. Aus diesem Grund ist es empfehlenswert OTRS über den Installer nur auf solchen Windowssystemen zu installieren, auf denen noch kein apache2 bzw. ein anderer Webserver und kein MySQL läuft.

Manuelle Installation (Linux, Unix)

Vorbereiten der manuellen Installation

Wenn Sie OTRS manuell über die Quellen installieren möchten oder müssen, laden Sie sich zuerst das aktuelle Archiv herunter. Sie finden die entsprechenden .tar.gz- oder .tar.bz2-Dateien im Downloadbereich auf <http://www.otrs.org/downloads> [<http://www.otrs.org/downloads>].

Entpacken Sie das Archiv mit Hilfe von **tar** z. B. in das Verzeichnis `/opt` und benennen Sie das entpackte Verzeichnis von 'otrs-3.1.x' in 'otrs' um:

```
linux:/opt# tar xf /tmp/otrs-3.1.tar.gz
linux:/opt# mv otrs-3.1 otrs
linux:/opt# ls
otrs
linux:/opt#
```

Skript: Erste Schritte bei der Installation von OTRS.

Da die Skripte von OTRS später nicht mit root-Rechten laufen sollen, muss im nächsten Schritt ein Benutzer für OTRS im System angelegt werden. Dieser Benutzer sollte als Homeverzeichnis das Verzeichnis erhalten, in das gerade die Quellen von OTRS entpackt wurden, also `/opt/otrs`. Wird der Webserver unter einem anderen Benutzer als dem OTRS-User betrieben, so muss der neue OTRS-Benutzer noch zur Gruppe des Webserver-Users hinzugefügt werden.

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

Skript: Anlegen eines neuen Nutzers 'otrs' und Hinzufügung zu desselben zu einer Gruppe.

Im nächsten Schritt werden einige Demo-Konfigurationsdateien innerhalb der entpackten Quelldateien bzw. innerhalb des Homeverzeichnisses des OTRS-Benutzers kopiert. Die Dateien befinden sich in den Verzeichnissen `/opt/otrs/Kernel` bzw. `/opt/otrs/Kernel/Config` und haben die Endung `.dist`.

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

Skript: Kopieren von Beispieldateien.

Zum Abschluss der Vorbereitungen werden noch die richtigen Zugriffsrechte für die Dateien des Ticket Systems gesetzt. Dazu kann das Skript **SetPermissions.sh** verwendet werden, das sich im Verzeichnis `bin` innerhalb des Homeverzeichnisses des OTRS-Benutzers befindet. Das Skript kann mit folgenden Parametern aufgerufen werden:

```
otrs.SetPermissions.pl {Homedirectory des OTRS Benutzers} {OTRS Benutzer}
{Webserver Benutzer} [Gruppe des OTRS Benutzers] [Gruppe des Webserver
Benutzers]
```

Läuft Ihr Webserver mit den Benutzerrechten des Benutzers 'otrs', dann lautet das Kommando also **otrs.SetPermissions.pl /opt/otrs --otrs-user=otrs --web-user=otrs**. Unter SUSE Linux wird der Webserver mit dem Benutzer `wwwrun` betrieben. Auf Debian-basierten Systemen ist dies `www-data`. Geben Sie hier das Kommando **otrs.SetPermissions.pl /opt/otrs --otrs-user=otrs --web-user=wwwrun --otrs-group=nogroup --web-group=www** ein, um die korrekten Zugriffsrechte zu setzen.

Installation der für OTRS benötigten Perl-Module

Für den Betrieb von OTRS werden einige Perl-Module benötigt. Wenn Sie OTRS manuell einrichten, müssen Sie wahrscheinlich einige dieser Module per Hand nachinstallieren. Dies können Sie entweder über den Paketmanager ihrer Distribution erledigen (yast, apt-get), oder, wie in diesem Kapitel beschrieben, direkt über die Shell von Perl und CPAN. Wir empfehlen die Verwendung des Paketmanagers, wenn möglich.

Tabelle 3.1. Die folgenden Perl-Module werden von OTRS benötigt.

Name	Beschreibung
DBI	Dieses Modul wird von OTRS für die Verbindung zum Datenbank-Backend benötigt.
DBD::mysql	Modul zum Verbindungsaufbau zum MySQL-Datenbank-Backend (nur benötigt bei Verwendung von MySQL).
DBD::pg	Modul zum Verbindungsaufbau zum PostgreSQL-Datenbank-Backend (nur benötigt bei Verwendung von PostgreSQL).
Digest::MD5	Ermöglicht die Verwendung des md5-Algorithmus.
CSS::Minifier	Komprimiert eine CSS-Datei und schreibt das Ergebnis in eine andere Datei.
Crypt::PasswdMD5	Stellt MD5-basierte kryptographische Funktionen bereit.
MIME::Base64	Kodiert / dekodiert Zeichenketten in Base64, z. B. für E-Mail-Anhänge.
JavaScript::Minifier	Komprimiert eine JavaScript-Datei und schreibt das Ergebnis in eine andere Datei.
Net::DNS	Perl-Interface zum DNS-System.
LWP::UserAgent	Verarbeitet HTTP-Anfragen.
Net::LDAP	LDAP-Schnittstelle für Perl (nur benötigt, wenn LDAP verwendet wird).
GD	Schnittstelle zur Gd Graphics Library. Wird nur benötigt, wenn das Statistikmodul von OTRS verwendet werden soll.
GD::Text, GD::Graph, GD::Graph::lines, GD::Text::Align	Text- und Grafikwerkzeuge für die Benutzung zusammen mit der GD Graphics Library. Diese Komponenten werden nur benötigt, wenn das Statistikmodul von OTRS verwendet werden soll.
PDF::API2, Compress::Zlib	Diese Module werden für die Generierung der PDF-Ausgaben von Statistiken, Suchergebnissen oder Druckansicht eines Tickets benötigt.

Sie können herausfinden, welche Module noch zu installieren sind, indem Sie das Skript **otrs.checkModules** verwenden. Das Skript finden sie im Verzeichnis `bin` innerhalb des Homeverzeichnisses des Benutzers 'otrs'.

Beachten Sie, dass einige Module optional sind.

```
linux:~# cd /opt/otrs/bin/
linux:/opt/otrs/bin# ./otrs.CheckModules.pl
  o CGI.....ok (v3.49)
  o Crypt::PasswdMD5.....ok (v1.3)
  o CSS::Minifier.....ok (v0.01)
  o Date::Format.....ok (v2.24)
  o Date::Pcalc.....ok (v1.2)
  o DBI.....ok (v1.609)
  o DBD::mysql.....ok (v4.013)
  o Digest::MD5.....ok (v2.36_01)
  o Encode::HanExtra.....ok (v0.23)
  o GD.....ok (v2.44)
    o GD::Text.....ok (v0.86)
    o GD::Graph.....ok (v1.44)
    o GD::Graph::lines.....ok (v1.15)
    o GD::Text::Align.....ok (v1.18)
  o IO::Scalar.....ok (v2.110)
  o IO::Wrap.....ok (v2.110)
  o JavaScript::Minifier.....ok (v1.05)
  o JSON.....ok (v2.21)
    o JSON::PP.....ok (v2.27003)
    o JSON::XS.....Not installed! (Optional - Install
it for faster AJAX/JavaScript handling.)
  o LWP::UserAgent.....ok (v5.829)
  o Mail::Internet.....ok (v2.06)
  o Mail::POP3Client.....ok (v2.18 )
    o IO::Socket::SSL.....ok (v1.31)
  o MIME::Base64.....ok (v3.07_01)
  o MIME::Tools.....ok (v5.428)
  o Net::DNS.....ok (v0.65)
  o Net::POP3.....ok (v2.29)
  o Net::IMAP::Simple.....ok (v1.1916)
    o Net::IMAP::Simple::SSL.....ok (v1.3)
  o Net::SMTP.....ok (v2.31)
    o Authen::SASL.....ok (v2.15)
    o Net::SMTP::SSL.....ok (v1.01)
  o Net::LDAP.....ok (v0.4001)
  o PDF::API2.....ok (v0.73)
    o Compress::Zlib.....ok (v2.008)
  o SOAP::Lite.....ok (v0.712)
  o Text::CSV.....ok (v1.18)
    o Text::CSV_PP.....ok (v1.26)
    o Text::CSV_XS.....Not installed! (Optional -
Optional, install it for faster CSV handling.)
    o XML::Parser.....ok (v2.36)
linux:/opt/otrs/bin#
```

Skript: Überprüfung benötigter Module.

Wenn möglich, sollten Sie die fehlenden Module über das Paketsystem Ihrer Linux-Distribution installieren. Dadurch werden die Pakete automatisch aktualisiert, wenn neue Versionen verfügbar sind oder Sicherheitslücken geschlossen wurden. Bitte schauen Sie dazu in die Dokumentation Ihrer Distribution. Wenn ihr Paket nicht (in der korrekten Version) im Paketsystem nicht verfügbar ist, können Sie es über das CPAN installieren, das Comprehensive Perl Archive Network.

Um eines der oben aufgeführten Module mit Hilfe von CPAN zu installieren, geben Sie als root das Kommando **perl -e shell -MCPAN** ein. Perl wird im interaktiven Modus gestartet und das CPAN Modul wird geladen. Ist CPAN bereits ordentlich konfiguriert, können Sie die für OTRS benötigten Module mit Hilfe des Kommandos **install** gefolgt vom Modulnamen einrichten. CPAN weist darauf hin, wenn Abhängigkeiten zwischen einzelnen Modulen nicht erfüllt sind und schlägt automatisch die zusätzlich benötigten Module für die Installation vor.

Führen Sie weiterhin die beiden Befehle **perl -cw bin/cgi-bin/index.pl perl -cw bin/cgi-bin/customer.pl** und **perl -cw bin/PostMaster.pl** aus, nach dem Sie in das Verzeichnis `/opt/otrs` gewechselt sind. Wird bei beiden Befehlen die Meldung "syntax OK" angezeigt, verfügt Ihre Perl-Installation über alle von OTRS benötigten Module und Sie können im nächsten Schritt mit der Einrichtung des Webserver beginnen.

```
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#
```

Skript: Crontab-Datei.

Konfiguration des Apache-Webserver

In diesem Abschnitt wird beschrieben, wie der Apache-Webserver grundlegend für OTRS eingerichtet werden muss. Der Webserver sollte cgi- bzw. Perl-Skripte ausführen können, andernfalls ist kein Betrieb von OTRS möglich. Überprüfen Sie die Konfigurationsdateien Ihres Webserver und stellen Sie fest, ob das cgi-Modul geladen wird: Wenn Ihr Webserver die Ausführung von cgi-Skripten unterstützt, sollte eine Zeile ähnlich der folgenden zu finden sein.

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

Um die Oberfläche von OTRS bequem erreichen zu können, wird ein Alias- und ein ScriptAlias-Eintrag angelegt. Für die meisten Installationen des apache Webserver gilt, dass ein Verzeichnis mit dem Namen `conf.d` vorhanden ist, unter Linux ist es meist unterhalb des Verzeichnisses `/etc/apache` bzw. `/etc/apache2` zu finden. Wechseln Sie als root in dieses Verzeichnis und kopieren Sie die passende Konfigurationsvorlage in `/opt/otrs/scripts/apache2-httpd.include.conf` in die Datei `otrs.conf`.

Starten Sie Ihren Webserver neu, um die neue Konfiguration zu laden. Auf den meisten Systemen lässt sich der Webserver über den Befehl **/etc/init.d/apache2 restart** neu starten (siehe skript unten).

```
linux:/etc/apache2/conf.d# /etc/init.d/apache2 restart
```

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

Skript: Neustart des Webservers.

Das war bereits die grundlegende Konfiguration des Webservers für OTRS.

Zur Verbesserung der Performance können sie `mod_cgi` deaktivieren und den Apache-Webserver stattdessen für die Benutzung von `mod_perl` konfigurieren.

Stellen Sie bitte sicher, daß `mod_perl` installiert und geladen ist. Andernfalls wird der Webserver-Dienst trotzdem starten, in diesem Fall wird jedoch `mod_cgi` verwendet, wenn es noch aktiviert ist (Deaktivierung wird empfohlen).

Suchen Sie im Verzeichnis `/etc/apache*` nach `mod_perl.so` (siehe Skript unten) um herauszufinden, ob es bereits geladen wird, oder nicht.

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

Skript: Suche nach mod_perl.

Wenn Sie das o. a. geeignete Startskript verwenden, und `mod_perl` geladen wird, kann das Skript `/opt/otrs/scripts/apache2-perl-startup.pl` verwendet werden, um die Perl-Module bereits beim Start des Webservers einmalig vorzuladen. Damit werden die Antwortzeiten kürzer, was die Performance insgesamt verbessert.

Einrichten der Datenbank

Der einfache Weg - Einrichtung der Datenbank mit Hilfe des Webinstallers (nur für MySQL)

Wenn Sie MySQL als Datenbank-Backend verwenden, können Sie den Web-Installer von OTRS verwenden: <http://localhost/otrs/installer.pl> [<http://localhost/otrs/installer.pl>] .

Der Webinstaller wird gestartet. Folgen Sie den Anweisungen auf dem Bildschirm.

1. Lesen Sie die Informationen über die OTRS-Niederlassungen und klicken Sie auf "Weiter".

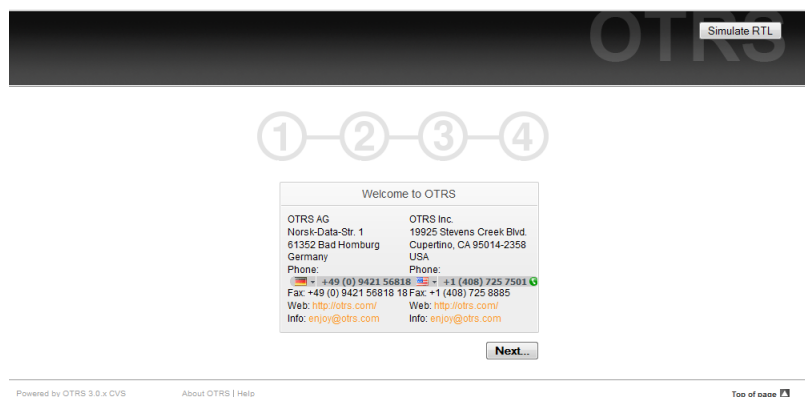


Abb.: Willkommensbildschirm.

2. Lesen Sie die GNU Affero General Public License und akzeptieren Sie sie durch Klick auf "Akzeptieren".



Abb.: GNU Affero General Public License.

3. Geben Sie Benutzernamen und Passwort für den Administrator an, den DNS-Namen des Servers, auf dem OTRS läuft, und den zu verwendenden Datenbanktyp. Prüfen Sie danach die Einstellungen.

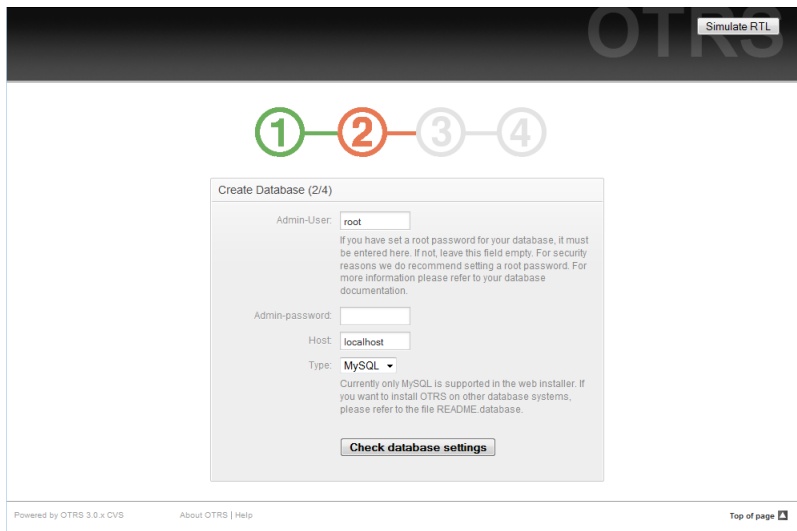


Abb.: Datenbank-Grundeinstellungen.

Wenn die Prüfung erfolgreich war, erhalten Sie einen Hinweis. Drücken Sie "Ok" zum Fortfahren.

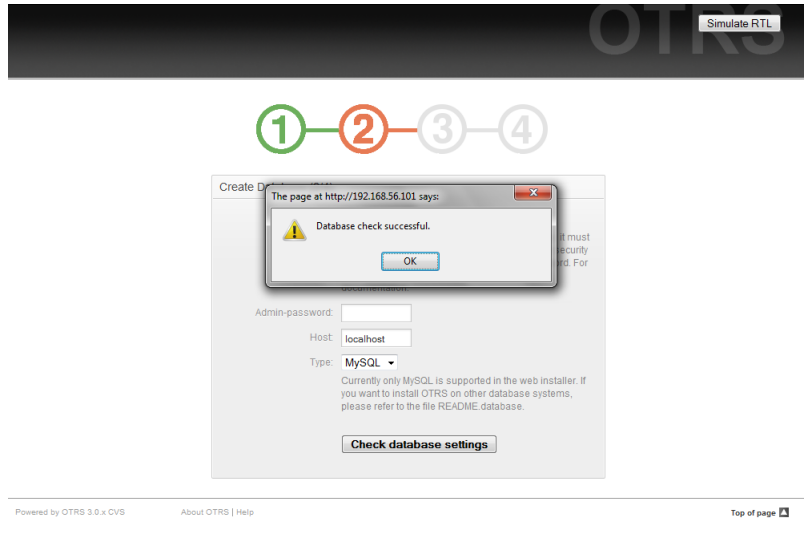


Abb.: Hinweis auf erfolgreiche Prüfung.

4. Erstellen Sie einen neuen Datenbankbenutzer, wählen Sie einen Namen für die Datenbank und klicken Sie auf "Weiter".

Warnung

Man sollte nie Standard-Passwörter verwenden. Bitte ändern Sie das Standardpasswort der OTRS-Datenbank!

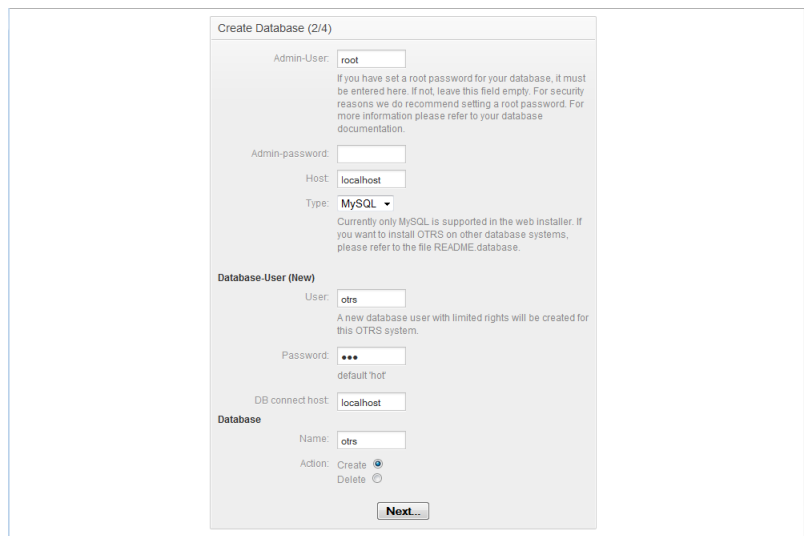


Abb.: Datenbankeinstellungen.

Wenn der Datenbankbenutzer und die Datenbank erfolgreich angelegt wurden, erhalten Sie eine Bestätigung. Klicken Sie auf 'Weiter', um fortzufahren.

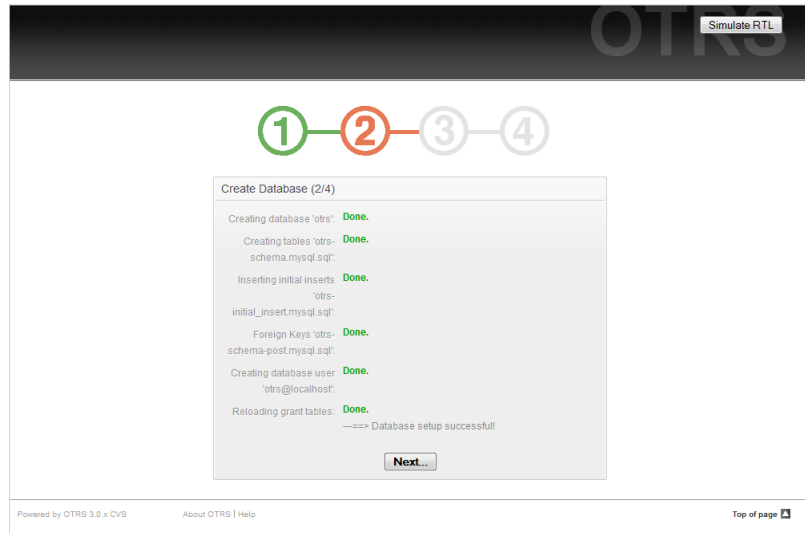


Abb.: Bestätigung der erfolgreichen Datenbankeinrichtung.

5. Geben Sie alle benötigten Systemeinstellungen an und klicken Sie "Weiter".

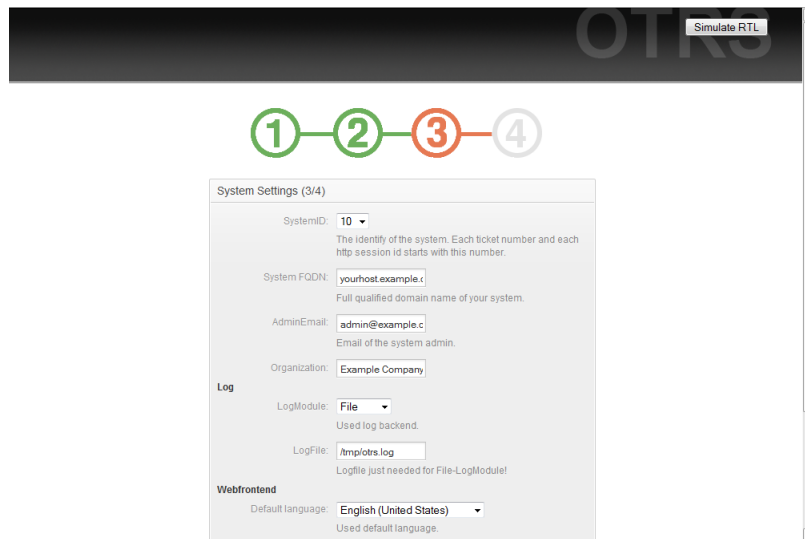


Abb.: Systemeinstellungen.

↓6. Wenn Sie wollen, können Sie an dieser Stelle eingehende und ausgehende E-Mail-Verbindungen einrichten, oder diesen Schritt überspringen.

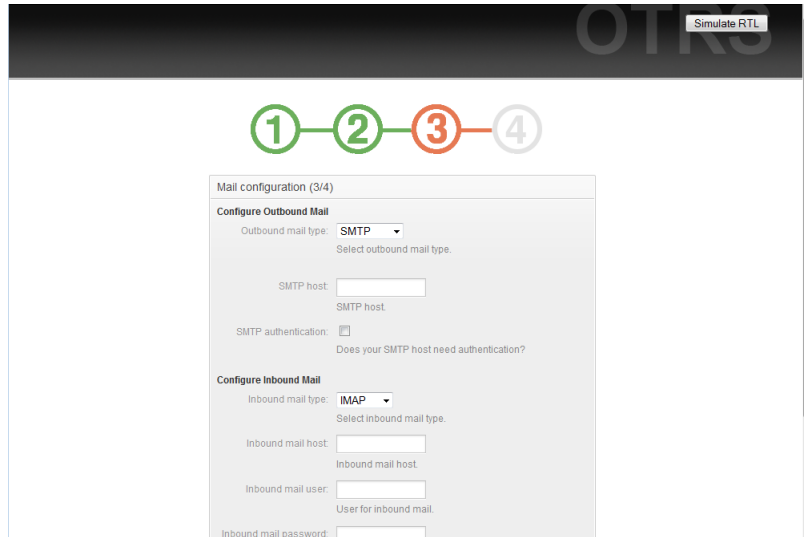


Abb.: E-Mail-Konfiguration.

7. Starten Sie den OTRS-Dienst neu, damit die neuen Konfigurationseinstellungen verwendet werden.

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

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

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

done

done
linux:~ #
```

Skript: Neustart des OTRS-Dienstes.

Herzlichen Glückwunsch! Die Installation ist abgeschlossen, jetzt können Sie mit dem System arbeiten. Um sich anzumelden, verwenden Sie bitte die Adresse `http://localhost/otrs/index.pl` [`http://localhost/otrs/index.pl`] in ihrem Webbrowser. Melden Sie sich als Administrator an, mit dem Benutzernamen `'root@localhost'` und dem Passwort `'root'`. Anschließend können Sie das System gemäß Ihrer Anforderungen konfigurieren.

Warnung

Bitte ändern Sie das Passwort für den Benutzer `'root@localhost'` umgehend.

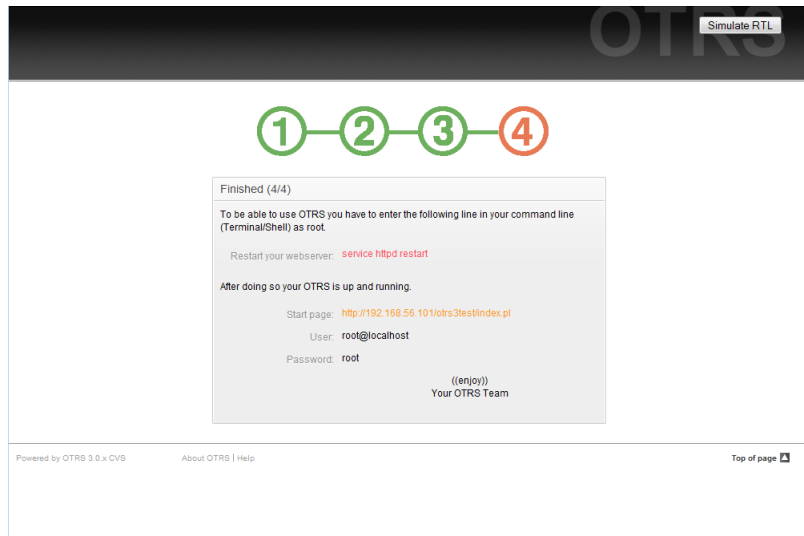


Abb.: Letzte Schritte zur Installation von OTRS.

Manuelle Installation der OTRS Datenbank

Wenn Sie den Webinstaller nicht einsetzen können, kann die Datenbank für OTRS auch manuell eingerichtet werden. Skripte mit dem Datenbankschema und den SQL-Statements finden Sie im Verzeichnis `scripts/database` innerhalb des Homeverzeichnisses des Benutzers `'otrs'`.

```
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#
```

Skript: Für die Datenbankerstellung benötigte Dateien.

Für die verschiedenen Datenbanktypen sind mehrere `.sql`-Dateien vorhanden, die nacheinander abgearbeitet werden müssen.

Die OTRS-Datenbank manuell Schritt für Schritt anlegen

1. Anlegen der Datenbank für OTRS: Legen Sie mit Hilfe Ihres Datenbankinterfaces bzw. Ihrer Datenbankoberfläche die Datenbank an, die später von OTRS verwendet werden soll.
2. Erstellen der Tabellen: Mit Hilfe der `otrs-schema.Datenbanktyp.sql`-Dateien (z. B. `otrs-schema.oracle.sql`, `otrs-schema.postgresql.sql`, usw.) können Sie die Tabellen innerhalb der OTRS Datenbank erzeugen.
3. Inserting the initial system data: OTRS needs some initial system data to work properly (e.g. the different ticket states, ticket and notification types). Depending on the type of your database, use one of the files `otrs-initial_insert.mysql.sql`, `otrs-initial_insert.oracle.sql`, `otrs-initial_insert.postgresql.sql` or `otrs-initial_insert.mssql.sql`.
4. Erzeugen von "foreign keys" auf andere Tabellen: Abschließend müssen noch die "foreign keys" erstellt werden, über die die verschiedenen Tabellen in der OTRS Datenbank voneinander abhängen. Dies kann mit Hilfe der `otrs-schema-post.Datenbanktyp.sql`-Dateien erreicht werden (z. B. `otrs-schema-oracle.post.sql`, `otrs-schema-post.postgresql.sql`, usw.).

Nachdem Sie die Datenbank angelegt haben, sollten Sie die Zugriffsrechte dafür setzen und z. B. sicherstellen, dass nur ein bestimmter Benutzer ohne Datenbank-Administrationsrechte Zugriff auf die OTRS Datenbank hat. Je nachdem, welche Datenbank Sie einsetzen, unterscheiden sich hier die Vorgehensweisen, es sollte jedoch möglich sein dies mit Hilfe Ihres Datenbankinterfaces bzw. Ihrer Datenbankoberfläche zu erledigen.

Wurden die nötigen Einstellungen für die Datenbank vorgenommen, muss nun noch dem Ticket-System mitgeteilt werden, welche Datenbank es verwenden soll. Öffnen Sie die Datei `Kernel/Config.pm` innerhalb des Homeverzeichnisses des OTRS Benutzers und passen Sie die dafür vorgesehenen Parameter an.

```
# 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';
```

Skript: Anzupassende Einstellungen.

Einrichten der von OTRS benötigten cron-Jobs

Damit OTRS voll funktioniert, werden einige cron-Jobs benötigt. Die cron-Jobs sollten mit denselben Benutzerrechten ausgeführt werden, die auch für die restlichen OTRS-Skripte vergeben wurden, d.h. die cron-Jobs sollten in die crontab des Benutzers 'otrs' eingetragen werden.

Alle Skripte für die verschiedenen cron-Jobs befinden sich im Verzeichnis `var/cron` innerhalb des Homeverzeichnis des Benutzers 'otrs'.

```
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#
```

Skript: Für die Erstellung der Cronjobs benötigte Dateien.

Alle Skripte tragen die Endung `.dist` und sollten zunächst so umkopiert werden, dass keine Endung mehr vorhanden ist.

```
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#
```

Skript: Kopieren und Umbenennen der für die Erstellung der Cronjobs benötigten Dateien.

Die folgende Tabelle gibt eine kurze Übersicht über die Aufgabe der verschiedenen Skripte.

Tabelle 3.2. Die verschiedenen Skripte für die cron-Jobs von OTRS.

Skript	Funktion
aaa_base	Über dieses Skript werden die grundlegenden Einstellungen für die crontab des Benutzers 'otrs' festgelegt.
cache	Löscht abgelaufene Cache-Einträge von der Festplatte. Leert den Cache von CSS- und JavaScript-Dateien.

Skript	Funktion
fetchmail	Falls Nachrichten mit Hilfe von fetchmail in das System eingespeist werden sollen, kann dieses Skript verwendet werden.
generic_agent	Mit Hilfe dieses Skripts werden die Jobs des GenericAgents ausgeführt, die über eigene Konfigurationsdateien festgelegt wurden.
generic_agent-database	Mit Hilfe dieses Skripts werden die Jobs des GenericAgents ausgeführt, die über dem Administrations-Bereich innerhalb von "GenericAgent" angelegt wurden.
pending_jobs	Mit Hilfe dieses Skripts wird das System auf "wartende" (pending) Tickets überprüft.
postmaster	Mit Hilfe dieses Skripts wird die Nachrichten-Warteschlange von OTRS überprüft und noch nicht verarbeitete Nachrichten werden im System gespeichert bzw. zugestellt.
postmaster_mailbox	Mit Hilfe dieses Skripts werden die verschiedenen pop3-Konten abgefragt, die im Administrations-Bereich innerhalb von "PostMaster POP3 Account" eingerichtet wurden.
rebuild_ticket_index	Mit Hilfe dieses Skripts wird der Ticket-Index für die Queue-Ansicht neu erzeugt, wodurch die Anzeige beschleunigt wird.
session	Über dieses Skript werden alte und nicht mehr gültige Session-IDs entfernt.
unlock	Mit Hilfe dieses Skripts wird die Freigabe von Tickets innerhalb des Systems ermöglicht.

Für die Einrichtung aller cron-Jobs kann das Skript `bin/Cron.sh` verwendet werden, das sich im Homeverzeichnis des OTRS-Benutzers befindet. Dem Skript muss beim Aufruf ein Parameter übergeben werden. Dieser Parameter legt fest, ob die cron-Jobs installiert, deinstalliert oder neu gestartet werden. Es sind folgende Parameter zulässig:

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

Da die cron-Jobs für den Benutzer 'otrs' angelegt werden sollen, muss das Skript von diesem Benutzer ausgeführt werden. Sind Sie z. B. als Benutzer root am System angemeldet, können Sie mit Hilfe des Kommandos **su otrs** zum OTRS-Benutzer wechseln. Nehmen Sie also die Installation wie folgt vor.

Warnung

Bitte beachten Sie, dass durch die Verwendung von `Cron.sh` evtl. andere Cron-Jobs des OTRS-Benutzers überschrieben bzw. gelöscht werden. Um weitere, nicht von OTRS benötigte Cron-Jobs für den OTRS-Benutzer zu installieren, erweitern Sie bitte `Cron.sh` dementsprechend.

```
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#
```

Skript: Installation der Cronjobs.

Mit Hilfe des Kommandos **crontab -l -u otrs**, das Sie als root ausführen können, wird die crontab-Datei des Benutzers 'otrs' angezeigt und Sie können überprüfen, ob alle Einträge vorhanden sind.

```
linux:/opt/otrs/bin# crontab -l -u otrs
# --
# cron/aaa_base - base crontab package
# Copyright (C) 2001-2012 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# Who gets the cron emails?
MAILTO="root@localhost"

# --
# cron/cache - delete expired cache
# Copyright (C) 2001-2012 OTRS AG, http://otrs.org/
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# delete expired cache weekly (Sunday mornings)
20 0 * * 0 $HOME/bin/otrs.CacheDelete.pl --expired >> /dev/null
30 0 * * 0 $HOME/bin/otrs.LoaderCache.pl -o delete >> /dev/null

# --
# cron/fetchmail - fetchmail cron of the OTRS
# Copyright (C) 2001-2012 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# fetch every 5 minutes emails via fetchmail
*/5 * * * * /usr/bin/fetchmail -a >> /dev/null

# --
# cron/generic_agent - otrs.GenericAgent.pl cron of the OTRS
# Copyright (C) 2001-2012 OTRS AG, http://otrs.org/
# --
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# start generic agent every 20 minutes
*/20 * * * * $HOME/bin/GenericAgent.pl >> /dev/null
# example to execute GenericAgent.pl on 23:00 with
```

```
# Kernel::Config::GenericAgentMove job file
#0 23 * * * $HOME/bin/otrs.GenericAgent.pl -c
  "Kernel::Config::GenericAgentMove" >> /dev/null
# --
# cron/generic_agent - GenericAgent.pl cron of the OTRS
# Copyright (C) 2001-2012 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# start generic agent every 10 minutes
*/10 * * * * $HOME/bin/otrs.GenericAgent.pl -c db >> /dev/null
# --
# cron/pending_jobs - pending_jobs cron of the OTRS
# Copyright (C) 2001-2012 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# check every 120 min the pending jobs
45 */2 * * * $HOME/bin/otrs.PendingJobs.pl >> /dev/null
# --
# cron/postmaster - postmaster cron of the OTRS
# Copyright (C) 2001-2012 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# check daily the spool directory of OTRS
#10 0 * * * * test -e /etc/init.d/otrs & /etc/init.d/otrs cleanup
>> /dev/null; test -e /etc/rc.d/init.d/otrs && /etc/rc.d/init.d/otrs
cleanup >> /dev/null
10 0 * * * $HOME/bin/otrs.CleanUp.pl >> /dev/null
# --
# cron/postmaster_mailbox - postmaster_mailbox cron of the OTRS
# Copyright (C) 2001-2012 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# fetch emails every 10 minutes
*/10 * * * * $HOME/bin/otrs.PostMasterMailbox.pl >> /dev/null
# --
# cron/rebuild_ticket_index - rebuild ticket index for OTRS
# Copyright (C) 2001-2012 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# just every day
01 01 * * * $HOME/bin/otrs.RebuildTicketIndex.pl >> /dev/null

# --
# cron/session - delete old session ids of the OTRS
# Copyright (C) 2001-2012 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# delete every 120 minutes old/idle session ids
```

```
55 */2 * * * $HOME/bin/otrs.DeleteSessionIDs.pl --expired >> /dev/null

# --
# cron/unlock - unlock old locked ticket of the OTRS
# Copyright (C) 2001-2012 OTRS AG, http://otrs.org/
# --
# This software comes with ABSOLUTELY NO WARRANTY.
# --
# unlock every hour old locked tickets
35 * * * * $HOME/bin/otrs.UnlockTickets.pl --timeout >> /dev/null

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

Skript: Crontab-Datei.

Upgrade des OTRS Frameworks

These instructions are for people upgrading OTRS from version 3.0 to 3.1, and apply both for RPM and source code (tarball) upgrades.

If you are running a lower version of OTRS you have to follow the upgrade path to 3.0 first (1.1->1.2->1.3->2.0->2.1->2.2->2.3->2.4->3.0->3.1 ...)!

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

If you need to do a "patch level upgrade", which is an upgrade for instance from OTRS version 3.1.1 to 3.1.3, you should skip steps 8, 10 and 12-19.

Please note that for upgrades from 3.1.beta1 or 3.1.beta2, an additional step 20 is needed!

If you are using Microsoft SQL Server as the DBMS for OTRS, please refer to the manual, chapter "Upgrading Microsoft SQL Server Data Types" for instructions how to upgrade the data types used by OTRS (<http://doc.otrs.org/3.1/en/html/upgrading-mssql-datatypes.html>).

1. Stoppen Sie alle relevanten Dienste.

z. B. (hängt von eingesetzten Diensten ab):

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

2. Sichern Sie folgende Daten in \$OTRS_HOME (Standard: OTRS_HOME=/opt/otrs):

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

3. Sichern Sie die Datenbank.

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

5. Setup new system (optional)

Neues System aufsetzen (optional). Wenn möglich, installieren Sie zunächst auf einer separaten Testmaschine.

6. Installieren Sie das neue Release (tar oder RPM).

- Mit dem Tarball:

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

Alte Configdateien wiederherstellen.

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

- Per RPM:

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

In diesem Fall stellt das RPM-Update die alten Konfigurationsdateien automatisch wieder her.

7. Eigene Themes

Note: The OTRS themes between 3.0 and 3.1 are NOT compatible, so don't use your old themes!

Themes are located under `$OTRS_HOME/Kernel/Output/HTML/*/*.dtl` (default: `OTRS_HOME=/opt/otrs`).

8. Setzen Sie die Datei-Berechtigungen.

wenn der Tarball benutzt wird, führen Sie

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

aus, um die Berechtigungen zu setzen, die für Ihr System benötigt werden.

9. Wenden Sie die Datenbankänderungen an (Teil 1/2):

```
shell> cd /opt/otrs/
```

```
# MySQL:
shell> cat scripts/DBUpdate-to-3.1.mysql.sql | mysql -p -f -u root
otrs
# PostgreSQL 8.2+:
shell> cat scripts/DBUpdate-to-3.1.postgresql.sql | psql otrs
# PostgreSQL, older versions:
shell> cat scripts/DBUpdate-to-3.1.postgresql_before_8_2.sql | psql
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;
```

Führen Sie das Migrationsskript aus (als OTRS-Benutzer, nicht als root!):

Sie müssen das Migrationsskript ausführen, um die Daten auf die neue Datenbankstruktur zu übernehmen. Bitte führen Sie

```
shell> scripts/DBUpdate-to-3.1.pl
```

aus.

Datenbankänderungen anwenden (Teil 2/2):

```
# MySQL:
shell> cat scripts/DBUpdate-to-3.1-post.mysql.sql | mysql -p -f -u
root otrs
# PostgreSQL 8.2+:
shell> cat scripts/DBUpdate-to-3.1-post.postgresql.sql | psql otrs
# PostgreSQL, older versions:
shell> cat scripts/DBUpdate-to-3.1-post.postgresql_before_8_2.sql |
psql otrs
```

10. Systemkonfiguration aktualisieren und Caches löschen. Bitte führen Sie aus:

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

11. Update your web server configuration

Note: this applies only if you use the Apache web server together with `mod_perl2`, and do not use the configuration file directly from the OTRS installation directory (e. g. with a symlink from the Apache configuration directory).

Please add a new setting to the Apache configuration file for OTRS:

```
# set mod_perl2 option for generic interface
<Location /otrs/nph-genericinterface.pl>
    PerlOptions -ParseHeaders
</Location>
```

Please see the file `/opt/otrs/scripts/apache2-httpd.include.conf` for an example of where this new option needs to be added (inside the `<IfModule mod_perl.c>` block).

In this file, you will also note a new section on caching:

```
<IfModule mod_headers.c>
    <Directory "/opt/otrs/var/httpd/htdocs/skins/*/css-cache">
        <FilesMatch "\.(css|CSS)$">
            Header set Cache-Control "max-age=2592000 must-
revalidate"
        </FilesMatch>
    </Directory>

    <Directory "/opt/otrs/var/httpd/htdocs/js/js-cache">
        <FilesMatch "\.(js|JS)$">
            Header set Cache-Control "max-age=2592000 must-
revalidate"
        </FilesMatch>
    </Directory>
</IfModule>
```

Please activate this in your local installation too, and make sure that `mod_headers` is installed and active.

12. Starten Sie Ihre Dienste neu.

e. g. (depends on used services):

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

Now you can log into your system.

13. Installierte Pakete überprüfen

Prüfen Sie im Paketmanager, ob alle Pakete noch als korrekt installiert markiert sind oder eine Re-Installation oder ein Upgrade benötigen.

14. Check for encoding issues

OTRS 3.1 only allows UTF-8 as internal charset. Non-UTF-8 installations of OTRS must switch to UTF-8.

15. Escalation events

If you want to use the new escalation events in your system, you need to activate the corresponding GenericAgent job in Kernel/Config/GenericAgent.pm. Please look into Kernel/Config/GenericAgent.pm.dist for an example of how to do this.

16.Ticket event handlers

The Event name TicketFreeTextUpdate_\$Counter was renamed to TicketDynamicFieldUpdate_\$FieldName. If you have any custom event handlers for these events, please adapt them.

17.DynamicField user preferences module

If you had one or more active custom settings for "PreferencesGroups###Freetext", you need to adapt them to work with the new DynamicFields engine. The PrefKey setting must be changed to "UserDynamicField_DynamicField", where the part after the _ is the name of the dynamic field. Existing values would need to be renamed in the database as well.

18.Custom free field default value event handler

If you used the event handler Ticket::EventModulePost###TicketFreeFieldDefault (not active by default), you'll need to migrate its configuration to the new setting Ticket::EventModulePost###TicketDynamicFieldDefault.

The configuration of this is slightly different; where you had to specify a Counter indicating the TicketFreeText number previously, now you need to specify the name of the DynamicField (for migrated fields, this will be DynamicField_TicketFreeKey\$Counter and DynamicField_TicketFreeText\$Counter. You need two separate entries now if you want to set both the key and the text field.

19.FreeText/Time based ACLs

If you have any ACLs defined which involve freetext or freetime fields, you need to adjust these ACL definitions.

Please have a look at <http://doc.otrs.org/3.1/en/html/acl.html>. There you can find a list of all possible ACL settings. In general, you need to add the prefix "DynamicField_" to existing free field definitions, and you can add a new "DynamicField" section to the "Properties" list for situations when a ticket does not exist yet.

20.Database Upgrade During Beta Phase

This step is ONLY needed if you upgrade from 3.1.beta1 or 3.1.beta2! Please apply the required database changes as follows:

MySQL:

```
shell> cat scripts/DBUpdate-3.1.beta.mysql.sql | mysql -p -f -u  
root otrs
```

PostgreSQL 8.2+:

```
shell> cat scripts/DBUpdate-3.1.beta.postgresql.sql | psql otrs
```

PostgreSQL, older versions:

```
shell> cat scripts/DBUpdate-3.1.beta.postgresql_before_8_2.sql |  
psql otrs
```

21.Gut gemacht!

Upgrade mit dem Windows-Installer

Derzeit gibt es keinen automatischen Upgrademechanismus für OTRS-Installationen, die mit dem Windows-Installer erstellt wurden. Das Upgrade besteht im Wesentlichen aus einem Backup der Datenbank und des Filesystems, dem Deinstallieren von OTRS, der Installation der neuere Version, der Wiederherstellung der Datenbank und dem Ausführen der Upgrade-Prozedur.

Das Upgrade wird beschrieben in FAQ# 4200351 [<http://faq.otrs.org/otrs/public.pl?Action=PublicFAQ;ItemID=351>], und es gibt zudem ein informatives YouTube-Video [<http://www.youtube.com/watch?v=sf0R-reMTWc>] available.

Upgrading Microsoft SQL Server Data Types

Starting OTRS version 3.1, OTRS uses the *NVARCHAR* data type rather than *VARCHAR* or *TEXT*, to store textual data. This is because the *NVARCHAR* type has full support for Unicode, whereas the old data types store data in UCS-2 format, which is a sub-set of Unicode. Also, the *TEXT* data type is deprecated since *SQL Server 2005*. Due to this, starting with OTRS version 3.1, the minimal SQL Server version required for operation with OTRS is now *Microsoft SQL Server 2005*.

Because dropping and re-creating these indexes is a time-consuming operation, especially on large databases, please plan enough time for performing the upgrade. We would recommend that you perform the upgrade on a copy of the database prior to doing the actual conversion to test the upgrade procedure and to time how much time will be needed on your specific environment.

Please make sure that, before you start, there is enough space available on the database server. Make sure the free space on your database server is at least 2.5x the current size of the database.

Wichtig

This upgrade procedure will upgrade all fields of the mentioned data types to the new types. This procedure first removes any indexes and constraints in which these fields are referenced, upgrades the fields, and then adds the indexes and constraints back. It will do so on all tables found in the SQL Server database that OTRS uses. If you would have stored non-OTRS tables in the OTRS database, and these tables contain columns of the data types *VARCHAR* or *TEXT*, these will also be updated.

1. Open a Command Line on the OTRS server.
2. Change directory to the OTRS root directory. If you're using the default OTRS installer this would be C:\Program Files\OTRS\OTRS.
3. Run the following command:

```
shell> perl scripts/DUpdate-to-3.1.mssql-datatypes.pl
```

4. This will generate three scripts in the specified directory scripts\database\update. Run these scripts on the SQL Server database, via SQL Server Management Studio or isql.

Kapitel 4. Erste Schritte

Dieser Abschnitt soll einen ersten Überblick über die Funktionsweise von OTRS und den Aufbau der Weboberfläche des Systems geben. Es werden die Unterschiede zwischen Mitarbeitern (Agents), Kunden (Customer) und Administratoren erklärt. Anschließend wird die erste Anmeldung als OTRS-Administrator durchgeführt und erläutert, was mit Hilfe der Benutzereinstellungen für jeden Account innerhalb des Systems festgelegt werden kann.

Agenten-Weboberfläche

Die Mitarbeiter bearbeiten über die Weboberfläche des Systems die Anfragen der Kunden, erstellen neue Tickets für andere Mitarbeiter oder Kunden, legen Tickets über Telefongespräche mit Kunden an, schreiben FAQ-Einträge, bearbeiten Kundendaten usw.

Supposing your OTRS host is reachable via the URL `http://www.example.com` [`http://www.example.com/`], then the OTRS login screen can be reached by using the address `http://www.example.com/otrs/index.pl` [`http://www.example.com/otrs/index.pl`] in a web browser (see Figure below).

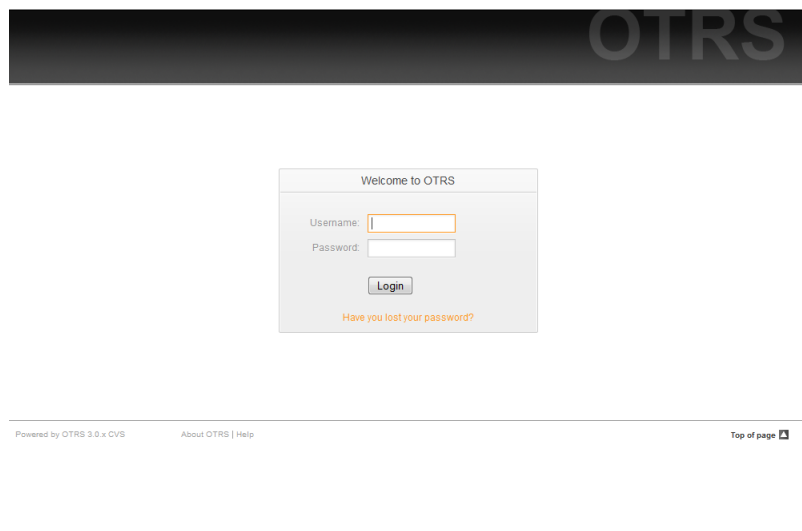


Abb.: Login-Bildschirm der Agenten-Weboberfläche.

Kunden-Weboberfläche

Kunden können über das speziell für sie vorhandene Webinterface von OTRS einen eigenen Kundenaccount anlegen, die eigenen Tickets einsehen, Tickets erstellen und bearbeiten, die Account-Einstellungen anpassen usw.

Gemäß dem obigen Beispiel kann der Login-Bildschirm für die Kunden über die URL `http://www.example.com/otrs/customer.pl` [`http://www.example.com/otrs/customer.pl`] geladen werden.

Company Support

Login

Username

Password

Login

Forgot password?

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

Powered by OTRS 3.0.x CVS

Abb.: Login-Bildschirm der Kunden-Weboberfläche.

Öffentliche Weboberfläche

Neben diesen beiden Bereichen der Weboberfläche verfügt OTRS weiterhin über ein Interface, welches vom FAQ-Modul bereit gestellt wird (muss separat installiert werden). Es ist öffentlich, also ohne Anmeldung, zugänglich und bietet Zugriff auf das FAQ-System.

Das öffentliche Web-Interface ist über die URL <http://www.example.com/otrs/faq.pl> [<http://www.example.com/otrs/faq.pl>] erreichbar. Ebenfalls ist ein Zugriff auf die FAQ über die URL <http://www.example.com/otrs/public.pl> [<http://www.example.com/otrs/public.pl>] möglich

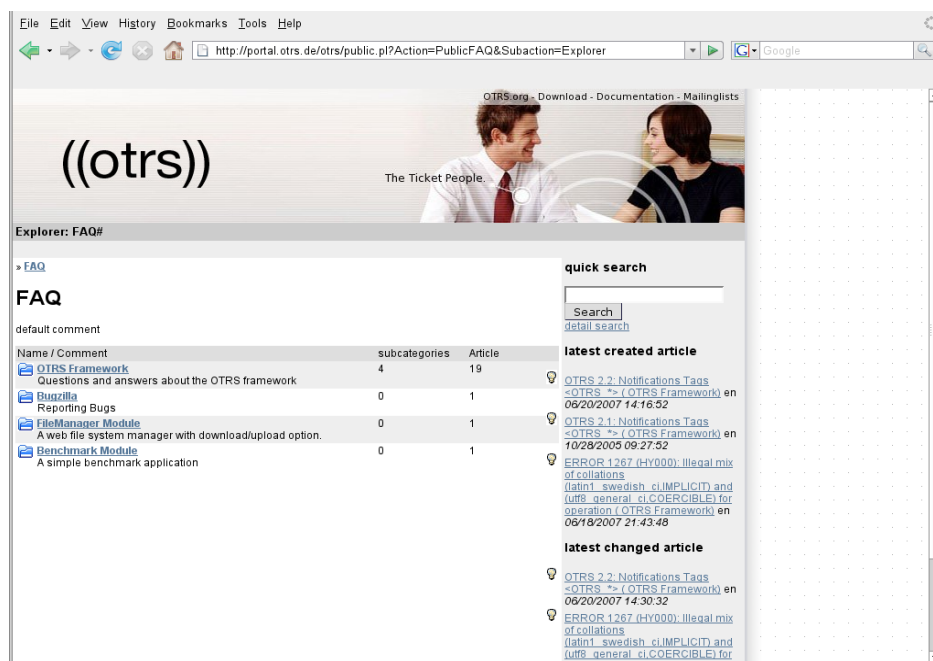


Abb.: Öffentliche Weboberfläche.

Die erste Anmeldung

Der Zugriff auf den Login-Bildschirm wird im Abschnitt [Agenten-Weboberfläche](#) beschrieben. Hier haben Sie die Möglichkeit, einen Benutzernamen und ein Kennwort anzugeben. Um sich als OTRS-Administrator anzumelden, verwenden Sie als Benutzername "root@localhost" und als Kennwort "root".

Warnung

Diese Zugangsdaten werden bei jeder OTRS-Installation standardmäßig vergeben. Da das Kennwort für den OTRS-Administrator somit öffentlich bekannt ist, sollten Sie es schnellstmöglich ändern! Sie können dies nach der Anmeldung als OTRS-Administrator über die Benutzereinstellungen vornehmen.

Wollen Sie sich nicht als OTRS-Administrator anmelden, geben Sie einfach den Benutzernamen und das Kennwort Ihres normalen OTRS-Accounts in die dafür vorgesehenen Eingabefelder ein.

Falls Sie einmal ihr Kennwort vergessen haben sollten, können Sie sich automatisch vom System ein neues Kennwort an die Mailadresse schicken lassen, die für Ihren OTRS-Account im System hinterlegt ist. Geben Sie dazu im unteren Bereich des Login-Bildschirms den Benutzernamen Ihres Accounts an.

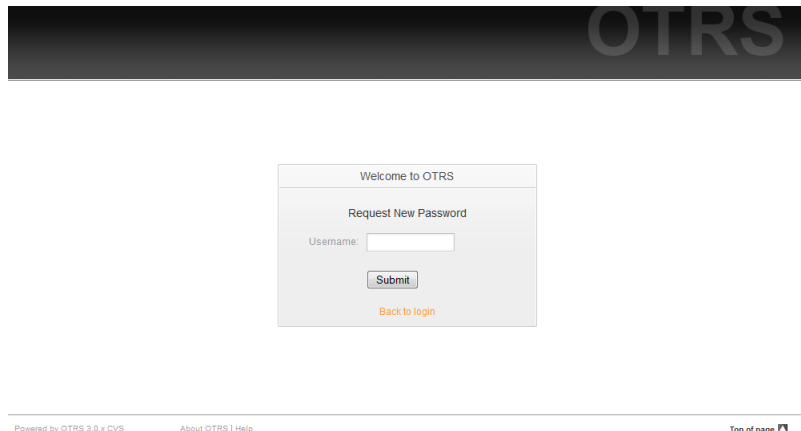


Abb.: Neues Passwort anfordern.

Agenten-Weboberfläche - ein Überblick

Nachdem Sie sich erfolgreich am System angemeldet haben, wird die Oberfläche von OTRS geladen. Standardmäßig befinden Sie sich nach der Anmeldung im Dashboard. Dieses Dashboard ist frei anpassbar. Zu Beginn zeigt es Ihre gesperrten Tickets und bietet Übersichten über fällige, eskalierte, neue und offene Tickets, neben anderen Dingen.

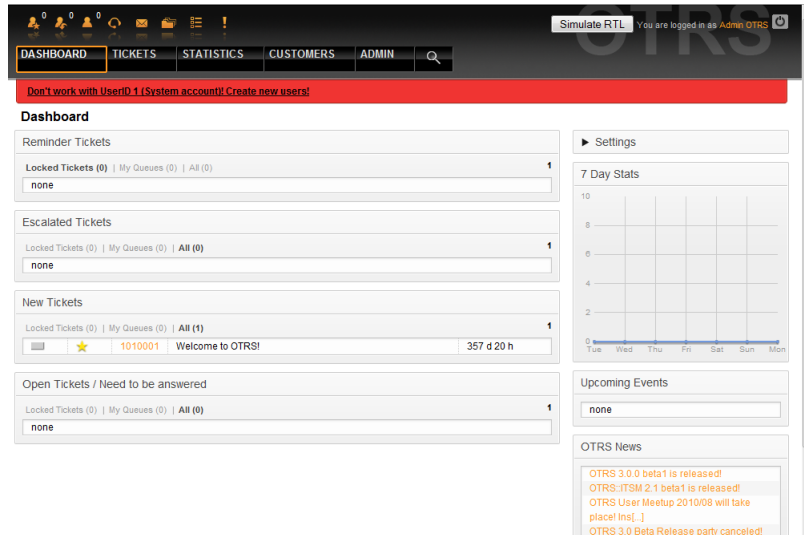


Abb.: Dashboard der Agenten-Weboberfläche.

Um die Übersichtlichkeit zu erhöhen, wurde die Oberfläche von OTRS in verschiedene Bereiche aufgeteilt. Oben finden sich einige allgemeine Informationen, wie der aktuelle Benutzername, der Abmelden-Knopf, und die Menge der gesperrten Tickets.

Unter diesen Icons befindet sich die Navigationsleiste. Sie zeigt Knöpfe, mit denen Sie die verschiedenen Bereiche und Module des Systems ansteuern und globale Aktionen auslösen können. Der Dashboard-Knopf führt zum Dashboard. Wenn sie den Ticket-Knopf drücken, erhalten Sie ein Untermenü mit verschiedenen Operationen, wie den Ticketansichten, der Erstellung neuer Tickets oder der Ticketsuche. Der Statistik-Knopf ist auch ein Menü, in dem Sie zwischen einer Übersicht der verfügbaren Statistiken, der Neuerstellung oder dem Import von Statistiken auswählen können. Der Kunden-Knopf führt in die Kunden-Verwaltung. Durch Druck auf den Admin-Knopf haben Sie Zugriff auf alle Administrator-Module, wo Sie z. B. neue Agenten, Queues usw. erstellen können. Zudem gibt es einen Suchknopf, mit dem man Suchvorgänge ausführen kann.

Wenn Zusatzmodule installiert werden, dann können diese weitere Menüpunkte zur Navigation hinzufügen.

Die rote Leiste unterhalb des Navigationsbereiches zeigt verschiedene Systemnachrichten an. Wenn Sie als OTRS-Administrator-Benutzer angemeldet sind, erhalten Sie hier den Hinweis, dass Sie mit diesem Benutzer nicht regulär arbeiten sollten.

Unter dem Haupttitel des Dashboards befinden sich verschiedene Unterbereiche, jeweils in einer eigenen Box. Diese können per Drag'n'Drop mit der Maus gezogen und losgelassen werden, um sie innerhalb ihrer Spalte neu anzuordnen. Dazu müssen Sie den Mauszeiger in den Kopfbereich der Boxen bewegen.

In der linken Spalte sehen Sie Daten über Tickets, unterteilt in: Erinnerung, eskaliert, neu und offen. In jeder Kategorie können Sie alle Tickets sehen, auf die Sie Rechte haben, wie viele Tickets Sie gesperrt haben und wie viele in "Meine Queues" liegen. "Meine Queues" sind Queues die Sie in Ihrer Benutzer-Konfiguration markieren können als Queues, die speziell überwacht werden sollen.

Auf der rechten Seite befindet sich der Bereich "Einstellungen". Klicken Sie darauf um den Bereich zu öffnen und die verschiedenen Einstellmöglichkeiten zu sehen. Sie können dann

Einstellungen verändern und Ihre Änderungen speichern. Dieser Bereich ist fixiert und kann nicht mit der Maus verschoben werden.

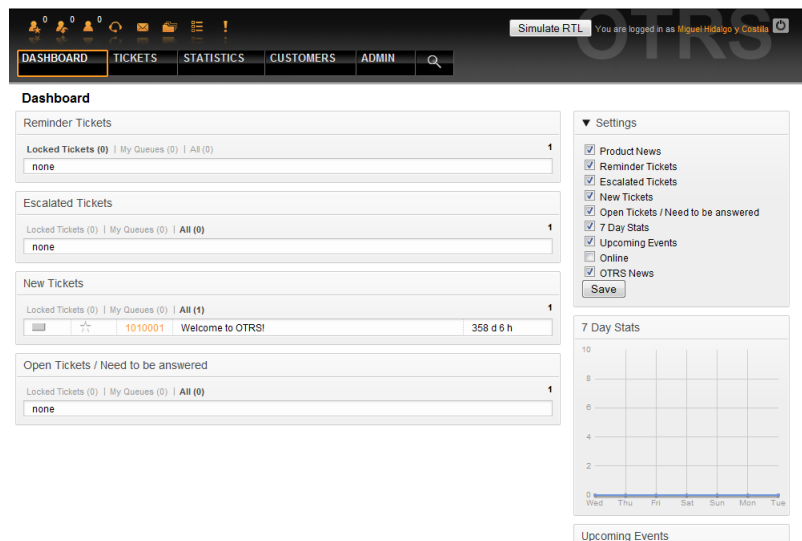


Abb.: Dashboard-Einstellungen.

Unter den Einstellungen sehen Sie die Statistik der Tickets der letzten 7 Tage. Zudem gibt es einen Abschnitt für anstehende Ereignisse und Neuigkeiten zu OTRS.

Im unteren Bildschirmbereich sehen sie die Fußzeile. Sie enthält u. A. Links auf die offizielle OTRS-Website und zum Seitenanfang.

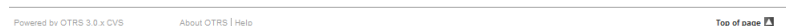


Abb.: Fußzeile.

Was verbirgt sich hinter dem Begriff Queue?

Normalerweise werden E-Mails in einer INBOX gespeichert und verwaltet. Eine INBOX ist eine große Datei, in der alle E-Mails aneinandergereiht werden. Neue E-Mails werden einfach an das Ende der INBOX angehängt. Das E-Mail-Programm, welches Sie zum Lesen und Bearbeiten Ihrer Nachrichten benutzen, liest die INBOX-Datei aus und bereitet den Inhalt für Sie als Nutzer auf.

Eine Queue in OTRS ist ein Mechanismus, mit dessen Hilfe viele Tickets gespeichert und verwaltet werden können, also auch eine Art INBOX. Als Anwender ist es völlig unwichtig zu wissen, wo oder wie das Ticket gesichert ist. Wichtig ist nur, zu wissen, welcher Queue das Ticket zugeordnet wurde. Anwender, also die sog. Agents (z. B. die Mitarbeiter ihrer Supportabteilung), können nun Tickets zwischen den Queues verschieben! Warum aber sollten sie das tun?

Gehen wir zur praktischeren Erklärung noch mal von Max Mustermanns Unternehmen aus dem Abschnitt ein Beispiel für ein Trouble Ticket System aus. Max Mustermann hat nach seinem anfänglichen Support-Chaos OTRS installiert, und er und seine Mitarbeiter nutzen das System zur Bearbeitung der Anfragen für die Videorekorder.

Eine Queue, in die alle Anfragen einsortiert werden, reicht in dieser Situation aus. Nach einiger Zeit bringt Max Mustermann einen DVD-Player auf den Markt, der von den Kunden gut

angenommen wird. Doch auch zu diesem Gerät laufen immer mehr Anfragen in das Ticket System und die Verwaltung der E-Mails mit einer Queue wird immer unübersichtlicher.

Deshalb entschließt sich Max Mustermann nach einiger Zeit, sein Supportsystem weiter zu optimieren. Er richtet zwei neue Queues ein, so dass er nun insgesamt drei Queues in OTRS definiert hat. Die erste und schon länger vorhandene Queue wird zur Eingangsqueue, in die erst mal alle Mails wandern, umfunktioniert. Daneben gibt es jetzt noch die neuen Queues "Videorekorder" und "DVD-Player".

Herr Mustermann beauftragt Frau Müller als sog. Dispatcherin tätig zu werden und mehrmals am Tag die Mails in der Eingangsqueue zu sichten und sie, je nach Inhalt, der Queue "Videorecorder" oder der Queue "DVD-Player" zuzuordnen. Herr Meier bearbeitet ab jetzt nur noch die Anfragen in der Videorekorder-Queue, Herr Schulze geht nur noch auf die Anfragen innerhalb der DVD-Player-Queue ein. Beide haben auf die jeweils anderen zwei Queues keinen Zugriff. Herr Mustermann kümmert sich weiter wie gewohnt um alle Arten von Anfragen und darf auf alle drei Queues zugreifen.

Weil OTRS Zugriffskontrolle für Agenten, Gruppen und Rollen unterstützt, ist es denkbar einfach, Queues einzurichten, die nur von bestimmten Agenten benutzt werden können. Herr Mustermann könnte auch ein anderes Szenario umsetzen, um seine Tickets in die verschiedenen Queues zu bekommen, mit Hilfe von Filterregeln, oder falls zwei verschiedene E-Mail-Adressen verwendet werden, müsste Frau Müller nur die E-Mails in die Queues zuordnen, die nicht automatisch einsortiert werden konnten.

Das Sortieren von Mails in verschiedene Queues schafft also Ordnung und mehr Übersicht in der täglichen Mailflut, deshalb sind Queues sehr wichtig für OTRS. Durch die Einteilung der Mitarbeiter (agents) in verschiedene Benutzergruppen mit differenzierten Zugriffsrechten auf die einzelnen Queues, kann die Abarbeitung der Anfragen weiter optimiert werden. Mit Hilfe von Queues können Sie die Struktur Ihres Unternehmens abbilden bzw. einzelne Geschäftsvorgänge abgrenzen. So könnte Max Mustermann neben seinem Support-Queues für die verschiedenen Geräte für Bestellungen eine Queue mit dem Namen "Sales" anlegen und als Unter-Queues "Anfragen", "Angebote", "Bestellungen" usw. definieren, um den Bestellvorgang zu optimieren.

Je besser und strukturierter ein Support- system organisiert ist, desto weniger Zeit und letztlich auch finanzielle Mittel müssen dafür aufgebracht werden. Queues und Unter-Queues helfen bei der Strukturierung bzw. bei der Abbildung von Abläufen.

Benutzereinstellungen

Die Einstellungen eines Accounts lassen sich mit Hilfe der Benutzereinstellungen den eigenen Wünschen entsprechend anpassen. Dabei spielt keine Rolle, ob man als Mitarbeiter, Kunde oder Administrator am System angemeldet ist. Die Benutzereinstellungen sind über den Link auf den Benutzernamen erreichbar, Kunden müssen den "Einstellungen"-Link verwenden.

Edit your preferences

User Profile

Change password

New Password:

Verify:

Password:

Language

Frontend language:

Spelling Dictionary

Default spelling dictionary:

Skin

Wear this frontend skin:

Theme

Email Settings

New ticket notification

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

Send new ticket notifications:

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:

Ticket lock timeout notification

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

Send ticket lock timeout notifications:

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:

QueueView Refresh Time

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

Refresh QueueView after:

Screen after new ticket

Show this screen after:

Abb.: Benutzereinstellungen des Agenten.

Ein Agent kann 3 verschiedene Kategorien von Einstellungen vornehmen: Benutzerprofil, E-Mail-Einstellungen und andere Einstellungen. Folgende Einstellungen sind möglich:

Benutzerprofil

- Aktuelles Passwort ändern.
- Sprache der Benutzeroberfläche wechseln.
- Oberflächen-Skin auswählen.
- Oberflächen-Theme auswählen.
- Abwesenheitszeit aktivieren und konfigurieren.

E-Mail-Einstellungen

- Ereignisse auswählen, bei denen eine E-Mail-Benachrichtigung durch das System erfolgen soll.

Andere Einstellungen

- Wählen Sie die Queues aus, die sie in "Meine Queues" überwachen wollen.
- Aktualisierungszeit der Queue-Ansicht einstellen.
- Bildschirm auswählen, der nach Erstellung eines Tickets angezeigt wird.

The screenshot displays the 'Company Personal Support' web interface. At the top, there is a header bar with the title 'Company Personal Support' and a 'Simulate RTL' button. Below the header is a navigation bar with links: 'New Ticket', 'My Tickets', 'Company Tickets', and 'Search'. On the right side of the navigation bar are 'Preferences' and 'Logout' links. The main content area is divided into several panels:

- Interface language:** A panel with a dropdown menu showing 'Your language' set to 'English (United States)' and an 'Update' button.
- Number of displayed tickets:** A panel with a dropdown menu showing 'Max. displayed tickets' set to '25' and an 'Update' button.
- Ticket overview:** A panel with a dropdown menu showing 'Refresh interval' set to 'off' and an 'Update' button.
- Change password:** A panel with two input fields, 'New Password' and 'Verify Password', and an 'Update' button.
- SMIME Certificate:** A panel with a text input field for 'SMIME Certificate Upload', a 'Browse...' button, and an 'Update' button.

At the bottom of the interface, there is a footer with the text 'Powered by OTRS 3.0.x CVS' and a link to 'About OTRS Help'.

Abb.: Benutzereinstellungen des Kunden.

Ein Kunde kann die Sprache der Benutzeroberfläche wechseln, die Aktualisierungszeit der Ticketübersichten verändern und die max. Anzahl angezeigter Tickets pro Seite einstellen. Zusätzlich kann das Passwort geändert werden.

Kapitel 5. Der Administrationsbereich von OTRS

Grundlagen

OTRS-Administratoren verwenden die Admin-Seite in der Weboberfläche von OTRS um das System zu konfigurieren - Agenten, Kunden und Queues hinzufügen, Ticket- und E-Mail-Einstellungen ändern und Zusatzpakete wie FAQ und ITSM installieren und vieles mehr ist damit möglich.

Agenten, die Mitglied der "admin"-Gruppe sind, können über den Link "Admin" innerhalb der Navigationsleiste der Agenten-Weboberfläche den Administrationsbereich von OTRS aufrufen.

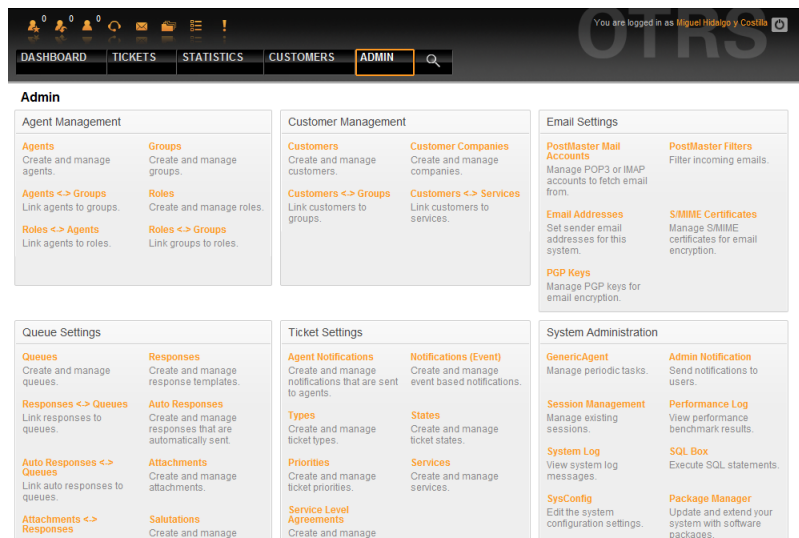


Abb.: OTRS Administrationsbereich.

Benutzer, Gruppen und Rollen

Benutzer

Über den Link "Benutzer" gelangen Sie in die Benutzerverwaltung von OTRS. Hier können Sie Benutzer anlegen, bearbeiten und deaktivieren. Weiterhin lassen sich einige grundlegende Einstellungen für den Benutzer festlegen, z. B. die Oberflächensprache oder das Anzeigeschema.

Anmerkung

Ein OTRS-Benutzer kann deaktiviert, aber nicht gelöscht werden. Um einen Benutzer zu deaktivieren, setzen Sie die Einstellung für "Gültig" auf "ungültig" oder "ungültig-temporär".

Der Administrationsbereich von OTRS

Agent Management

Actions

[Add agent](#)

Filter for Agents

Note

Agents will be needed to handle tickets.
Attention: Don't forget to add a new agent to groups and/or roles!

USERNAME	NAME	EMAIL	LAST LOGIN	VALID	CHANGED	CREATED
fcoaxavier.mina	Francisco Xavier Mina	fcoaxavier.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

Powered by OTRS 3.0.x CVS

Top of page

Abb.: Benutzerverwaltung.

Um einen neuen Benutzer anzulegen, klicken Sie auf den "Benutzer hinzufügen"-Knopf, geben Sie die benötigten Daten an und klicken Sie auf Absenden.

Agent Management

Actions

[Go to overview](#)

Note

Agents will be needed to handle tickets.
Attention: Don't forget to add a new agent to groups and/or roles!

Add Agent

Title:

Firstname:

Lastname:

Username:

Password:

Email:

Valid:

Language:

Spelling Dictionary:

Skin:

Theme:

Out Of Office Time: ☐ On ☒ Off

Start: 08 / 03 / 2010

Abb.: Hinzufügen eines Benutzers.

Nachdem Sie einen neuen Benutzer angelegt haben, muss dieser einer Gruppe bzw. einer Rolle zugewiesen werden. Sie werden nach dem Anlegen eines neuen Benutzers automatisch auf die Bildschirmmaske für die Zuweisung eines Benutzers in Gruppen weitergeleitet. Weitere Informationen über Gruppen und Rollen finden Sie in den Abschnitten Gruppen und Rollen dieses Kapitels.

Gruppen

Jeder Mitarbeiter mit einem Account im OTRS, sollte mindestens einer Benutzergruppe angehören. In einer Neuinstallation sind drei vordefinierte Gruppen verfügbar.

Tabelle 5.1. Standardmäßig vorhandene Gruppen in OTRS

Gruppe	Beschreibung
admin	Gruppe für die Benutzer mit Administrationsrechten.
stats	Benutzer in dieser Gruppe dürfen lesend (ro) oder schreibend (rw) auf das Statistikmodul von OTRS zugreifen, d.h. sie können Statistiken einsehen oder auch erstellen.
users	Dies ist die Gruppe, in die normale Mitarbeiter aufgenommen und mit den kompletten Rechten ausgestattet werden sollten. Dadurch wird für die Mitarbeiter das normale Arbeiten im System ermöglicht, der Zugriff auf alle Funktionen rund um Tickets ist gegeben.

Anmerkung

In einer neuen Installation ist die Gruppe "users" leer und der Benutzer "root@localhost" ist Mitglied der Gruppen "admin" und "stats".

Sie können durch Klick auf "Gruppen" im Administrationsbereich in die Gruppenverwaltung gelangen.

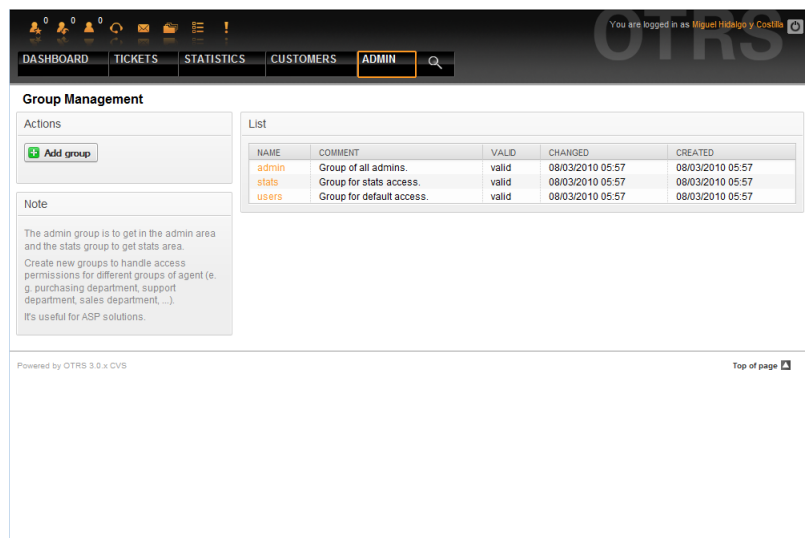


Abb.: Gruppenverwaltung.

Anmerkung

In OTRS können Gruppen deaktiviert, aber nicht gelöscht werden. Deaktivieren Sie eine Gruppe, indem Sie für "Gültig" den Wert entweder auf "ungültig" oder auf "ungültig-temporär" setzen.

Um einen Benutzer einer Gruppe zuzuweisen bzw. die Gruppenmitgliedschaft eines Benutzers zu ändern, kann der Link "Benutzer <-> Gruppen" im Administrationsbereich genutzt werden.

Der Administrationsbereich von OTRS

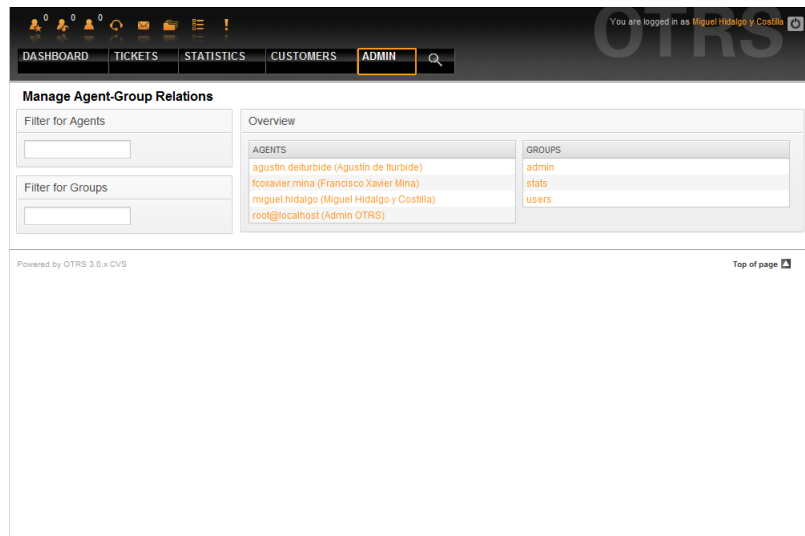


Abb.: Gruppenverwaltung.

Im unteren Bereich des Bildschirms wird eine Übersicht angezeigt, die alle Benutzer und Gruppen auflistet. Indem Sie auf einen Benutzernamen klicken, bekommen Sie dessen Gruppenzugehörigkeiten angezeigt und können diese ändern. Bei der Auswahl einer Gruppe werden alle Benutzer aufgelistet, die sich in dieser Gruppe befinden.

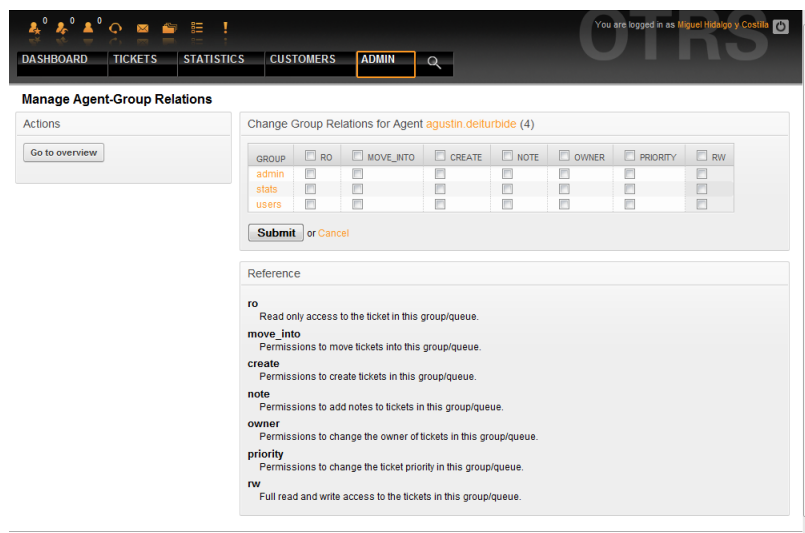


Abb.: Bearbeiten der Gruppenzugehörigkeiten eines Benutzers.

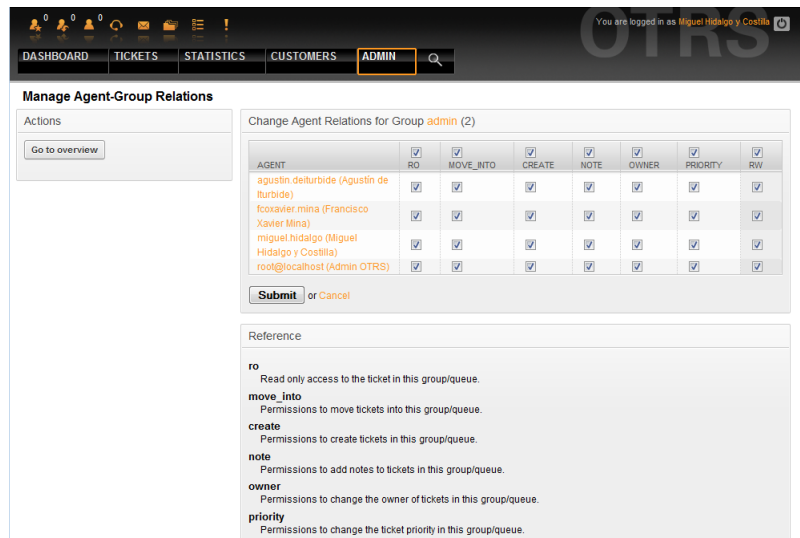


Abb.: Bearbeiten der Benutzerzuordnungen einer Gruppe.

Jede Gruppe hat zugehörige Berechtigungen, und jeder Agent, der Mitglied dieser Gruppe ist, kann eine Kombination dieser Rechte erhalten. Eine Liste dieser Berechtigungen wird in Tabelle 5-2 gezeigt.

Tabelle 5.2. Mit OTRS-Gruppen verknüpfte Rechte

Berechtigung	Beschreibung
ro	Nur Lesezugriff auf die Tickets bzw. Beiträge dieser Gruppe bzw. der Gruppe zugewiesenen Queues oder Bereiche.
Verschieben in (move into)	Recht zum Verschieben von Tickets oder Beiträgen innerhalb der Queues bzw. Bereiche dieser Gruppe.
erstellen	Recht zum Erstellen von Tickets oder Beiträgen in den Queues, bzw. Bereichen dieser Gruppe.
Besitzer (owner)	Recht zum Ändern des Eigentümers von Tickets, bzw. Beiträgen in den der Gruppe zugewiesenen Queues bzw. Bereiche.
Priorität	Recht zum Ändern der Priorität von Tickets, bzw. Beiträgen in den der Gruppe zugewiesenen Queues bzw. Bereiche.
rw	Voller Lese- und Schreibzugriff auf alle Inhalte der dieser Gruppe zugewiesenen Queues, bzw. Bereiche.

Anmerkung

Standardmäßig listet die Queue-Ansicht nur Tickets in Queues auf, auf welche der Benutzer *rw*-Rechte hat, d. h. Tickets, an denen der Benutzer arbeiten muss. Wenn Sie dieses Verhalten ändern wollen, können Sie `Ticket::Frontend::AgentTicketQueue###ViewAllPossibleTickets` auf *Ja* setzen.

Rollen

Rollen sind ein sehr nützliches und mächtiges Feature in OTRS, um schnell und einfach die Vergabe von Zugriffsrechten für viele Benutzer vorzunehmen. Vor allem bei großen und komplexen Installationen mit vielen Benutzern, Gruppen und Queues, zahlt sich dieses Feature schnell aus und erspart dem OTRS-Administrator viel Zeit und Arbeit. Das nachfolgende Beispiel stellt ein mögliches Szenario vor.

Um den Nutzen von Rollen zu verdeutlichen, stellen Sie sich die Situation vor, dass Sie ein OTRS-System mit 100 Benutzern verwalten. 90 Benutzer haben Zugriff auf eine Queue namens Support, die mehrere themenspezifische Unter-Queues enthält und in der die Support-Anfragen Ihrer Kunden landen. Die restlichen Queues des Systems sind für die 90 Supporter nicht zugänglich, dies wurde durch Gruppenzugriffsrechte so festgelegt. Die übrigen 10 Benutzer haben Zugriff auf alle Queues im System. Sie sortieren falsch einsortierte Mails aus, behalten die "Raw"-Queue im Auge und verschieben Spam-Mails in die "Junk"-Queue.

Im Rahmen einer Unternehmensumstrukturierung wird eines Tages zusätzlich eine Abteilung eröffnet, die Produkte verkaufen soll. Es müssen Angebote, Auftragsbestätigungen und Rechnungen erstellt, Anfragen bearbeitet, Bestellungen ans Lager weitergeleitet und Stornierungen entgegen genommen werden. Ein Teil der bisherigen Mitarbeiter soll in verschiedenen Bereichen der neuen Abteilung tätig werden und Sie als OTRS-Administrator haben nun die Aufgabe die neuen Queues anzulegen, die erweiterten Zugriffsrechte anzupassen und diese für die einzelnen Benutzer zu ändern.

Da es mühsam und viel zu umständlich wäre, für einen Teil aller 100 Benutzer einzeln die Zugriffsrechte zu ändern, richten Sie Rollen ein die mit Hilfe von Gruppenberechtigungen die verschiedenen Zugriffsrechte regeln. Anschließend ändern Sie für die entsprechenden Benutzer auf einmal die Zugriffsberechtigungen, indem Sie diese der entsprechenden Rolle zuweisen. Beim Anlegen neuer Benutzer müssen Sie nicht mehr einzeln die Gruppen und Zugriffsrechte einstellen, auch hier genügt die Verknüpfung des neuen Benutzers mit einer Rolle.

Anmerkung

Rollen sind wirklich hilfreich für die Verwaltung großer OTRS-Installationen. Allerdings sollten Sie nicht die Zuordnung von Agenten zu Gruppen und Rollen gleichzeitig einsetzen, um die Wartung der Zugriffsberechtigungen nicht zu kompliziert zu machen. Sollten Sie sich also entscheiden, auf Rollen zu setzen, empfehlen wir Ihnen, die Verwaltung für die Benutzer-Gruppen-Zuordnungen abzuschalten, indem Sie `Frontend::Module###AdminUserGroup` in der `SysConfig` deaktivieren. Dadurch werden existierende Gruppenzuordnungen natürlich nicht entfernt!

Sie können die Rollenverwaltung durch Klick auf "Rollen" im Administrationsbereich erreichen.

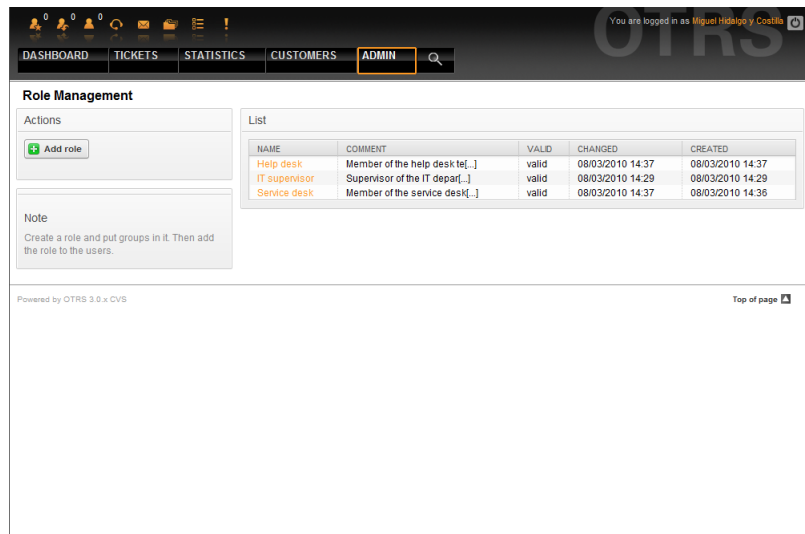


Abb.: Rollenverwaltung.

Anmerkung

In OTRS können Rollen deaktiviert, aber nicht gelöscht werden. Deaktivieren Sie eine Rolle, indem Sie für "Gültig" den Wert entweder auf "ungültig" bzw. "ungültig-temporär" setzen.

Eine Übersicht aller Rollen im System erscheint, um einen Eintrag zu bearbeiten, klicken Sie einfach auf den Rollennamen. In einer Neuinstallation gibt es keine vordefinierten Rollen. Um eine anzulegen, klicken Sie auf den "Rolle hinzufügen"-Knopf und geben Sie die erforderlichen Daten an.

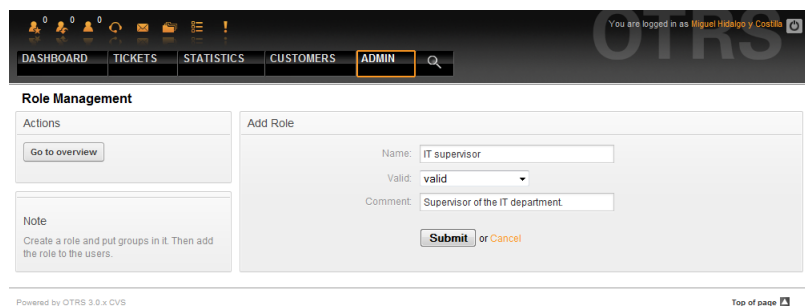


Abb.: Anlegen einer neuen Rolle.

Um einen Überblick über alle Rollen und Agenten im System zu erhalten, klicken Sie auf den Link "Rollen <-> Benutzer" im Administrationsbereich. Sie können die Filter verwenden, um einen bestimmten Eintrag zu finden. Wenn Sie die Rollenzordnungen eines Benutzers ändern wollen,

klicken Sie auf den Benutzernamen. Um die Benutzerzuordnungen einer Rolle zu verändern, klicken Sie auf die Rolle.

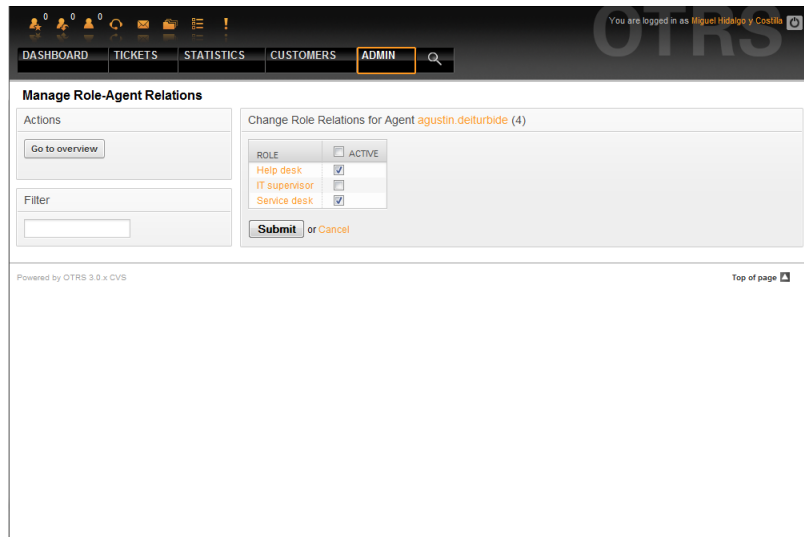


Abb.: Bearbeiten der Rollenzuordnungen eines Agenten.

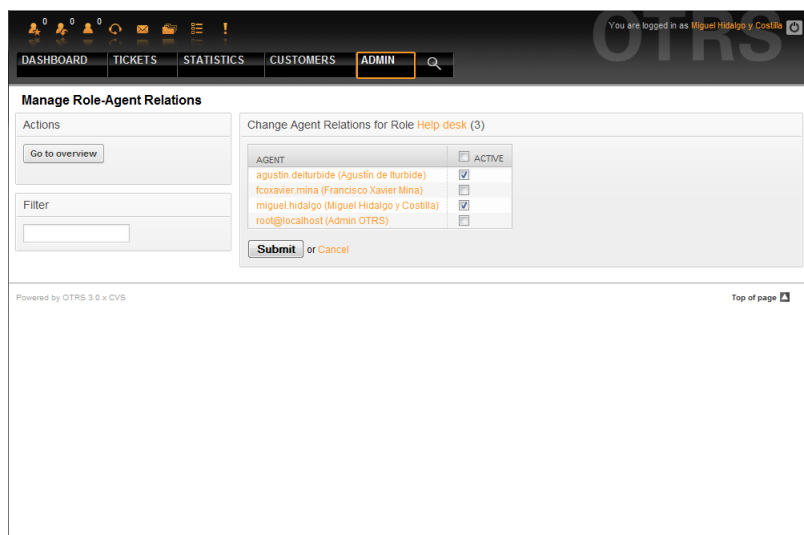


Abb.: Bearbeiten der Benutzerzuordnungen einer Rolle.

Um eine Übersicht aller Rollen und Gruppen im System zu erhalten, verwenden Sie den Link "Rollen <-> Gruppen" im Administrationsbereich.

Der Administrationsbereich von OTRS

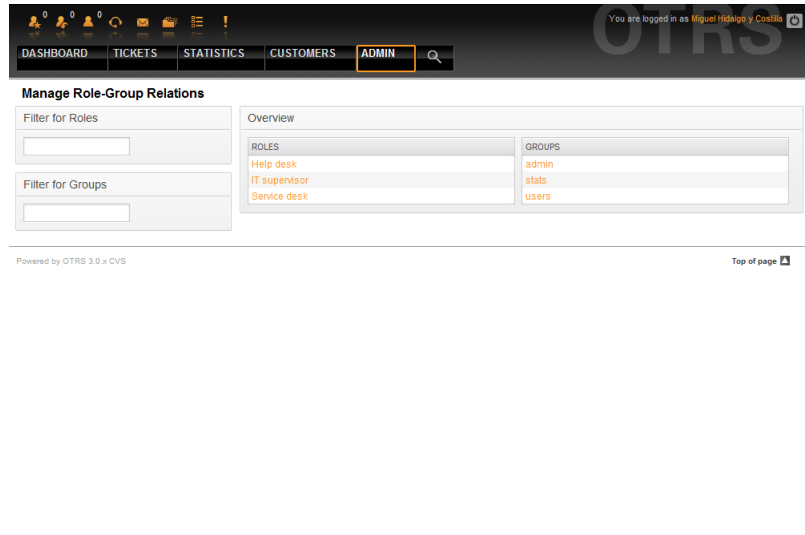


Abb.: Rollen-Gruppen-Zuordnungen verwalten.

Klicken Sie auf eine Gruppe oder Rolle!, um die jeweiligen Zuordnungen zu verändern.

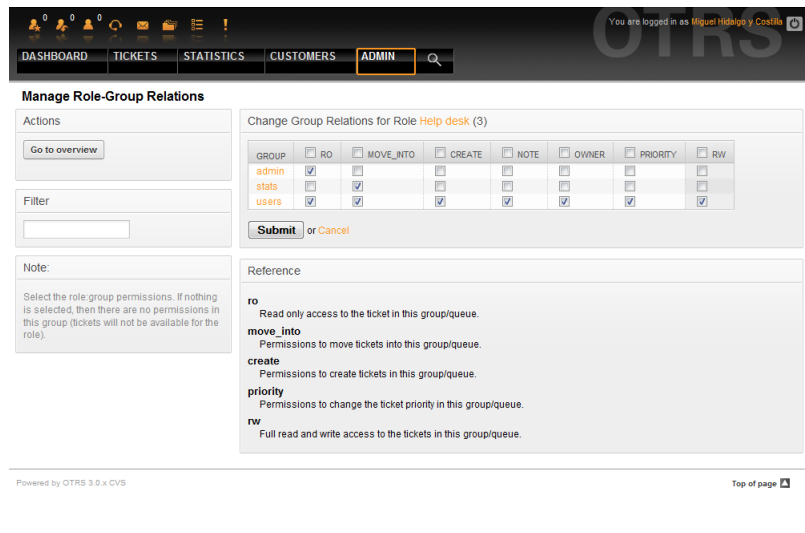


Abb.: Gruppenzuordnungen einer Rolle verändern.

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

Submit or Cancel

Reference

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

move_into
Permissions to move tickets into this group/queue.

create
Permissions to create tickets in this group/queue.

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

Abb.: Rollenzuordnungen für eine Gruppe verändern.

Kundenbenutzer und Kundengruppen

Kunden

OTRS unterstützt, wie bereits erwähnt, verschiedene Arten von Benutzern. Über den Link "Kunden", den Sie im Admin-Bereich von OTRS finden, können Sie die Benutzerdaten der im System angelegten Kunden verwalten. Ein Kunde kann sich mit Hilfe seines Accounts in das vom Ticket-System bereitgestellte Webinterface für Kunden einloggen, um dort die eigenen Tickets einzusehen, neue Tickets zu verfassen, usw. Weiterhin wird ein Kunden-Account vom System für die Historie von Tickets benötigt.

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

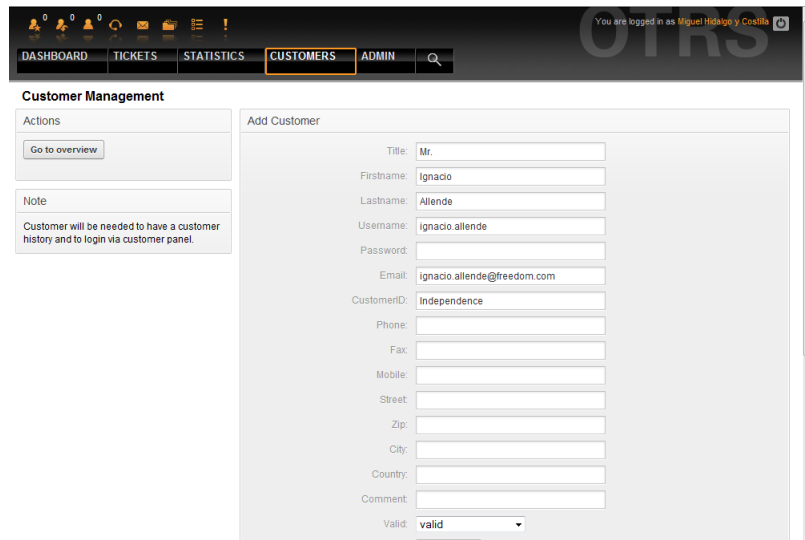
Powered by OTRS 3.0.x CVS

Top of page

Abb.: Kundenverwaltung.

Neben der Möglichkeit in der Datenbank nach einem bestimmten Kunden zu suchen, kann das Backend umgestellt werden, über das auf die Kundendaten zugegriffen wird. In OTRS lassen sich mehrere Datenbanken mit Kundendaten einbinden, genauere Informationen hierzu finden Sie im Abschnitt Einbinden externer Backends für Agents und Customer.

Um einen neuen Kunden hinzuzufügen, klicken Sie auf "Kunde hinzufügen". Einige Felder sind Pflichtfelder, müssen also ausgefüllt werden.



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

Abb.: Hinzufügen eines Kunden.

Der Kunde kann mit Benutzername und Passwort auf das System zugreifen. Die Kundennummer wird vom System benötigt, um den Kunden und seine Tickets zu erkennen. Da E-Mail-Adressen eindeutig sind, können sie als ID verwendet werden.

Anmerkung

In OTRS können Kunden deaktiviert, aber nicht gelöscht werden. Sie können einen Kunden deaktivieren, indem Sie den Wert für "Gültig" auf "ungültig" oder "ungültig-temporär" setzen.

Kundengruppen

Kunden können auch Gruppen zugeordnet werden. Das ist nützlich, wenn Kunden nur auf bestimmte Queues zugreifen können sollen. Legen Sie zuerst die benötigten Gruppen in der Gruppenverwaltung an. Fügen Sie dann die Queues hinzu und ordnen Sie diese der neuen Gruppe zu.

Im nächsten Schritt aktivieren Sie mit Hilfe des Konfigurationsparameters CustomerGroupSupport die Unterstützung für Kundengruppen. Mit Hilfe des Parameters CustomerGroupAlwaysGroups legen Sie fest, welchen Gruppen ein neu angelegter Kundenbenutzer automatisch zugeordnet werden soll.

Über den Link "Kunden <-> Gruppen" können Sie nun die Zuordnung der Kundenbenutzer in die gewünschten Gruppen vornehmen.

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

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

Abb.: Kundenzuordnungen einer Gruppe bearbeiten.

Queues

Über den Link "Queue" innerhalb des Admin-Bereiches von OTRS können Sie die Queues Ihres Systems verwalten. Nach einer Standard-Installation sind bereits die Queues "Junk", "Misc", "Postmaster" und "Raw" im System angelegt. "Raw" ist die Default-Queue, in ihr landen alle neuen Tickets, so lange kein Filter definiert wurde. "Junk" kann z. B. zum Aussortieren von Spam-Mails genutzt werden.

Abb.: Queue-Verwaltung.

Hier können Sie Queues anlegen und verändern. Zusätzlich zum Namen der neuen Queue kann angegeben werden, für welche Benutzergruppe die Queue bereitgestellt werden und ob die neue Queue eine Unter-Queue (sub queue) von einer bereits in Ihrem System vorhandenen Queue sein soll.

The screenshot shows the OTRS Admin interface. At the top, there's a navigation bar with tabs: DASHBOARD, TICKETS, STATISTICS, CUSTOMERS, and ADMIN (which is selected). Below the navigation bar, there's a 'Manage Queues' section. On the left, there's an 'Actions' box with a 'Go to overview' button. The main area is titled 'Add Queue'. It contains several input fields and dropdown menus: 'Name' (set to 'Support'), 'Sub-queue of' (a dropdown), 'Group' (set to 'admin'), 'Unlock timeout minutes' (set to '1220'), 'Escalation - first response time (minutes)' (set to '60'), 'Escalation - update time (minutes)' (set to '180'), and 'Escalation - solution time (minutes)' (set to '300'). Each time field has a '(Notify by: ...)' dropdown next to it. Below each time field, there is a small explanatory text: '0 = no escalation - 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.' for the unlock timeout; and '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.' for the escalation times.

Abb.: Hinzufügen einer Queue.

Wurde ein Ticket von einem Agenten gesperrt, so können Sie mit Hilfe des Freigabezeit-Intervalls festlegen, wann ein Ticket wieder automatisch vom System freigegeben werden soll. So können auch die anderen Mitarbeiter wieder auf dieses Ticket zugreifen und es bearbeiten.

Drei Arten von Eskalationszeiten können pro Queue eingestellt werden:

Eskalation - Zeit für die erste Reaktion

- Nach der Erstellung wird dieses Ticket eskalieren, wenn die hier definierte Zeit verstreicht, ohne dass eine Kommunikation zum Kunden (per E-Mail oder Telefonanruf) stattgefunden hat.

Eskalation - Aktualisierungszeit

- Wenn ein Artikel vom Kunden hinzugefügt wird, als E-Mail-Antwort oder über die Kundenoberfläche, wird die Aktualisierungszeit zurückgesetzt. Wenn anschließend für die definierte Zeitdauer keine Rückmeldung an den Kunden erfolgt, eskaliert das Ticket.

Eskalation - Lösungszeit

- Wenn das Ticket nicht innerhalb der definierten Zeitdauer geschlossen werden kann, eskaliert es.

Weiterhin können Sie festlegen, dass bei einem Follow-Up auf ein Ticket wieder der Mitarbeiter Eigentümer dieses Tickets wird, der zuletzt als Eigentümer im System für dieses Ticket vermerkt war. Dies stellt sicher, dass die Nachfrage eines Kunden zuerst bei demselben Mitarbeiter landet, der sich zuletzt um dieses Ticket gekümmert hat.

Der Parameter für die Systemadresse legt fest, mit welcher Absenderadresse Mails aus dieser Queue versendet werden sollen. Mit Hilfe der Parameter für Anrede und Signatur kann eingestellt werden, welche Vorgaben hier standardmäßig bei Antworten auf Tickets in dieser Queue genutzt werden. In den Abschnitten E-Mail-Adressen, Anreden und Signaturen erfahren Sie mehr über die Einrichtung dieser Parameter.

Anmerkung

In OTRS können Queues deaktiviert, aber nicht gelöscht werden. Sie können eine Queue deaktivieren, indem Sie den Wert für "Gültig" auf "ungültig" oder "ungültig-temporär" setzen.

Anreden, Signaturen, Anlagen und Antwortvorlagen

Anreden

Ein Textbaustein für eine Antwortvorlage ist die Anrede. Anreden können einer Queue zugeordnet werden, wie im Abschnitt zu den Queues beschrieben, nur so werden Sie bei Antworten für Tickets aus dieser Queue verwendet. Über den Link "Anreden" innerhalb des Admin-Bereiches, können Sie die verschiedenen Anreden Ihres Systems verwalten.

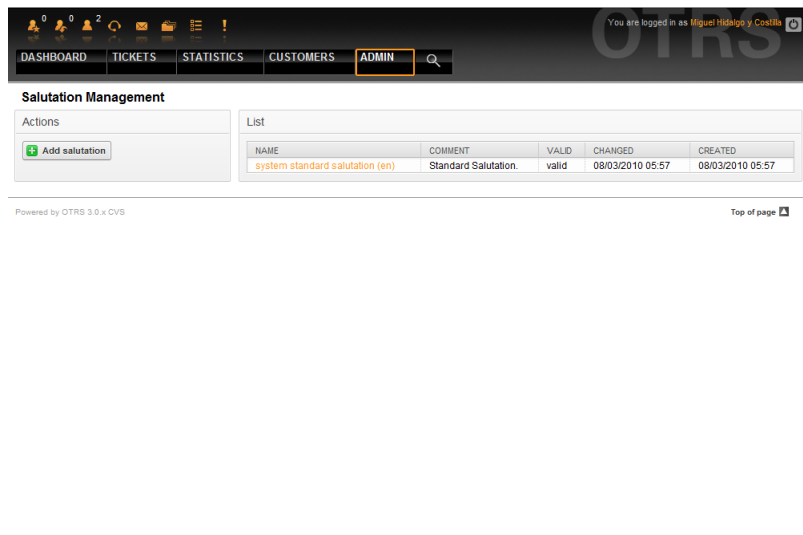


Abb.: Verwaltung der Anreden.

Nach einer Standardinstallation von OTRS sind bereits eine Anrede im System gespeichert, "system standard salutation (en)".

Um eine neue Anrede anzulegen, verwenden Sie den Knopf "Anrede hinzufügen" und geben Sie die erforderlichen Daten an.

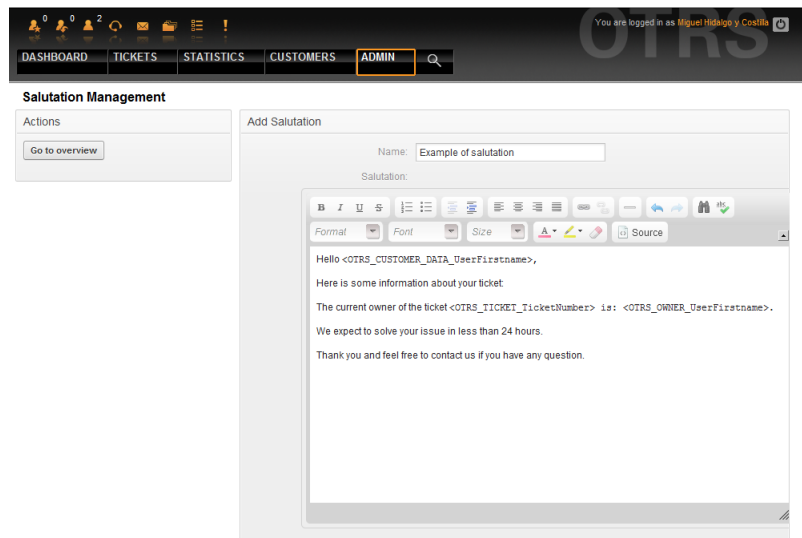


Abb.: Anrede hinzufügen.

In Anreden können Variablen verwendet werden. Wenn Sie auf ein Ticket antworten, werden die Variablennamen durch die zugehörigen Werte ersetzt.

Im unteren Teil der Bildschirmmaske zum Ändern oder Anlegen einer Anrede finden Sie die verschiedenen OTRS-Variablen, die Sie für die dynamische Aufbereitung eines Anredentextes verwenden können. Bauen Sie z. B. in Ihrem Anredentext die Variable `<OTRS_LAST_NAME>` ein, so wird diese später durch den Nachnamen der Person ersetzt, deren Ticket Sie beantworten.

Anmerkung

In OTRS können Anreden deaktiviert, aber nicht gelöscht werden. Sie können eine Anrede deaktivieren, indem Sie den Wert für "Gültig" auf "ungültig" oder "ungültig-temporär" setzen.

Signaturen

Ein weiterer Textbaustein für eine Antwortvorlage, ist die Signatur. Signaturen können einer Queue zugeordnet werden, wie im Abschnitt zu den Queues beschrieben, nur so werden Sie bei Antworten für Tickets aus dieser Queue verwendet. Über den Link "Signaturen" innerhalb des Admin-Bereiches, können Sie die verschiedenen Signaturen Ihres Systems verwalten.

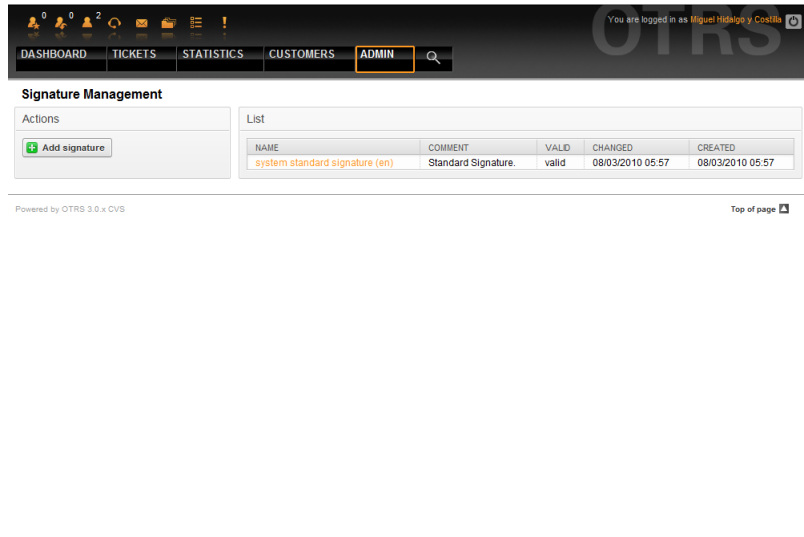


Abb.: Signaturverwaltung.

Nach einer Standardinstallation von OTRS ist bereits eine Signatur im System vorhanden, "system standard signature (en)".

Verwenden Sie den Knopf "Signatur hinzufügen", um eine neue Signatur anzulegen.

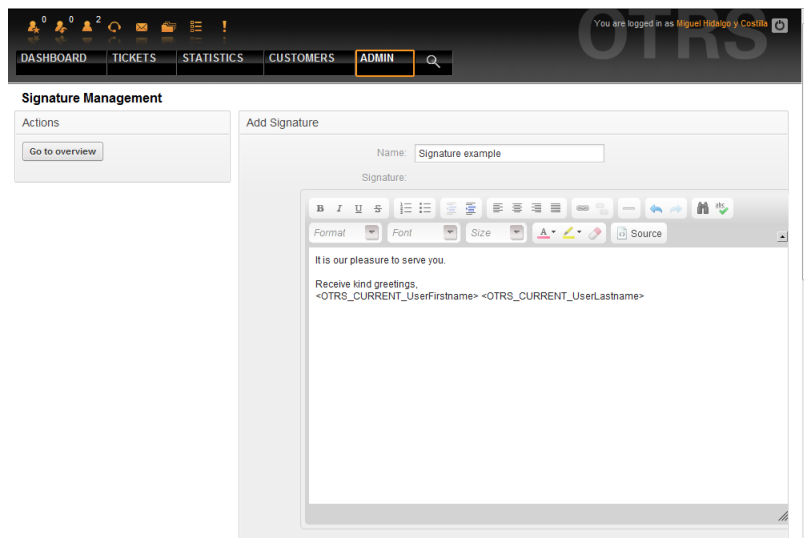


Abb.: Signatur hinzufügen.

Auch in Signaturen können dynamische Inhalte eingebaut werden, wie etwa Vor- und Nachname des Agenten. Dies geschieht, genauso wie bei den Anreden, mit Hilfe verschiedener OTRS-Variablen, die in den Text der Signatur integriert werden können. Ein variabler Textteil könnte z. B. der Name des Mitarbeiters sein, der das Ticket beantwortet hat (<OTRS_LAST_NAME>).

Anmerkung

In OTRS können Signaturen deaktiviert, aber nicht gelöscht werden. Sie können eine Signatur deaktivieren, indem Sie den Wert für "Gültig" auf "ungültig" oder "ungültig-temporär" setzen.

Anlagen

Ein weitere optionaler Teil einer Antwortvorlage kann eine Anlage sein. Diese wird bei der Benutzung einer Antwortvorlage als Anhang an das zu beantwortende Ticket angehängt, kann aber über eine Checkbox im Antworten-Bildschirm für Tickets leicht deaktiviert werden.

Über den "Anlagen"-Link im Admin-Bereich können Sie neue Anlagen in das System integrieren.

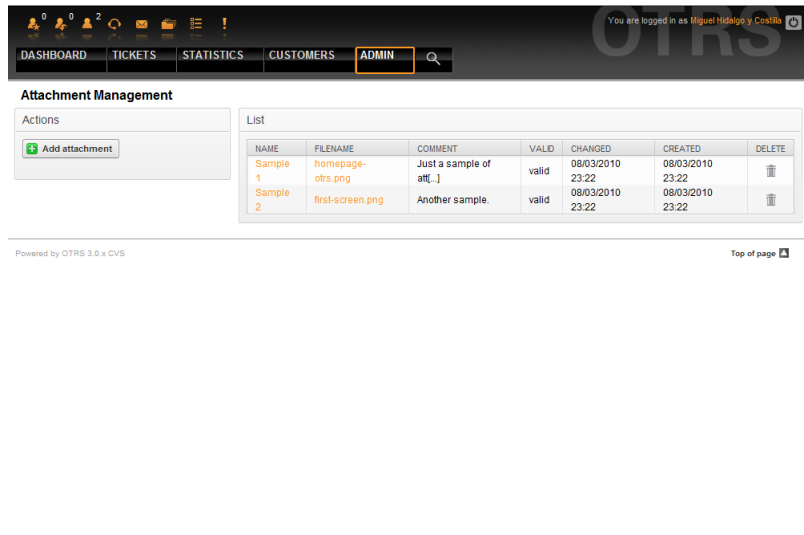


Abb.: Verwaltung der Anlagen.

Verwenden Sie den "Anlage hinzufügen"-Knopf, um eine neue Anlage zu erstellen.

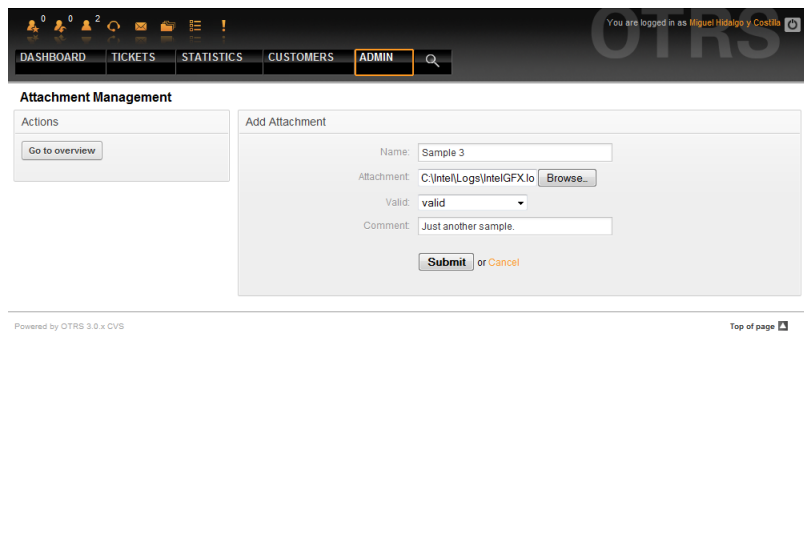


Abb.: Anlage hinzufügen.

Wenn eine Anlage erstellt wurde, kann sie einer oder mehreren Antworten zugeordnet werden. Folgen Sie dazu dem Link "Anlagen <-> Antworten" im Administrationsbereich.

Der Administrationsbereich von OTRS

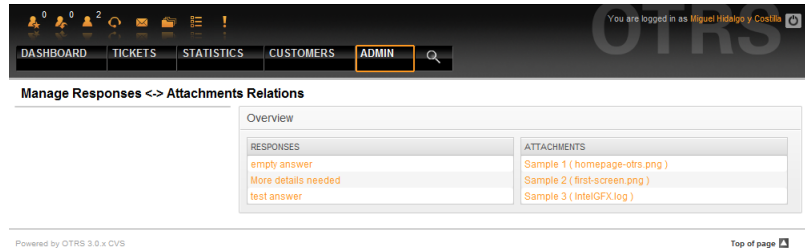


Abb.: Zuordnung von Anlagen zu Antworten.

Klicken Sie auf den Namen einer Anlage oder einer Antwortvorlage, um die jeweiligen Zuordnungen zu verändern.

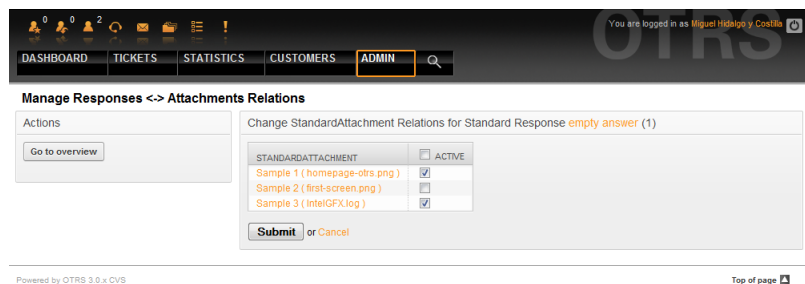


Abb.: Anlagenzuordnungen einer Antwortvorlage verändern.

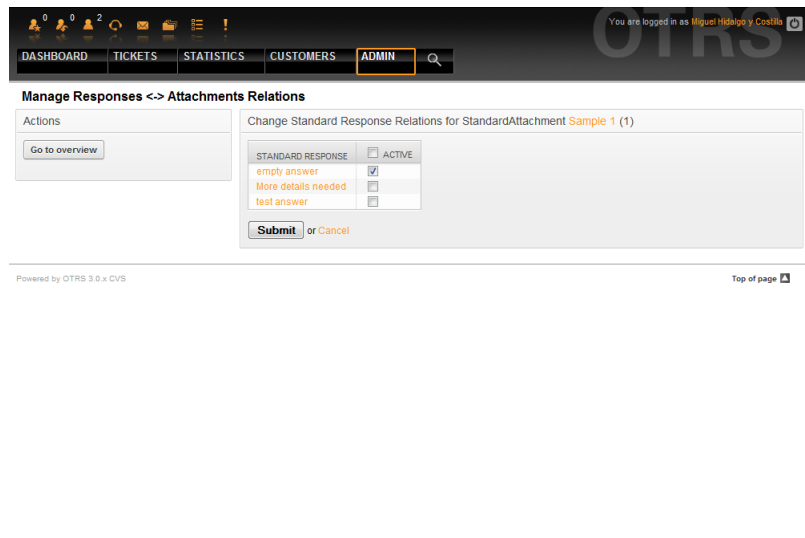


Abb.: Antwortvorlagen-Zuordnungen einer Anlage verändern.

Antwortvorlagen

Um das Antworten auf Tickets zu beschleunigen und das Aussehen beantworteter Tickets zu vereinheitlichen, können in OTRS Antwortvorlagen erstellt werden. Jede Antwortvorlage kann einer oder mehreren Queues bzw. Unter-Queues zugeordnet werden, es können mehrere Antwortvorlagen für jede Queue definiert werden.

Nach einer Standardinstallation ist für jede Queue die Antwortvorlage "empty answer" vorhanden. Im Admin-Bereich kann über den Link "Antworten" auf die im System gespeicherten Antwortvorlagen zugegriffen und diese bearbeitet werden.

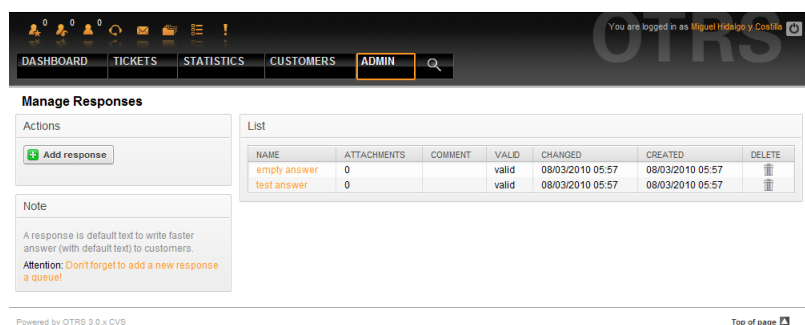


Abb.: Verwaltung der Antwortvorlagen.

Verwenden Sie den "Antwort hinzufügen"-Knopf zur Erstellung einer neuen Antwortvorlage.

Der Administrationsbereich von OTRS

The screenshot shows the OTRS Admin interface. The top navigation bar includes links for DASHBOARD, TICKETS, STATISTICS, CUSTOMERS, and ADMIN (which is highlighted). The user is logged in as 'Riguel Hidalgo y Costilla'. The main section is titled 'Manage Responses'. On the left, there are 'Actions' (Go to overview) and a 'Note' box with a warning: 'Attention: Don't forget to add a new response a queue!'. The main area is the 'Add Response' form. It has a 'Name' field with 'More details needed'. Below it is a 'Response' text area with a rich text editor toolbar. The text in the area is: 'Hello <OTRS_CUSTOMER_DATA_UserFirstname>, We haven't solved your issue, due to the lack of information previously provided. We need to know the following things: Greetings, <OTRS_OWNER_UserFirstname>'. The footer of the page indicates 'Powered by OTRS 3.0.x CVS' and a 'Top of page' link.

Abb.: Antwortvorlage hinzufügen.

Um eine Antwortvorlage einer oder mehreren Queues zuzuweisen, aktivieren Sie im Admin-Bereich den Link "Antworten <-> Queues". Sie können die Filter verwenden, um bestimmte Elemente zu finden.

The screenshot shows the 'Manage Response-Queue Relations' overview in the OTRS Admin interface. The top navigation bar is the same as the previous screenshot. The main section is titled 'Manage Response-Queue Relations'. On the left, there are two filter boxes: 'Filter for Responses' and 'Filter for Queues'. The main area is an 'Overview' table with two columns: 'RESPONSES' and 'QUEUES'. The 'RESPONSES' column lists 'empty answer', 'More details needed', and 'test answer'. The 'QUEUES' column lists 'Junk', 'Misc', 'Postmaster', 'Raw', and 'Support'. The footer of the page indicates 'Powered by OTRS 3.0.x CVS' and a 'Top of page' link.

Abb.: Queue-Antwort-Zuordnungen verwalten.

Klicken Sie auf den Namen einer Queue oder Antwortvorlage, um die jeweiligen Zuordnungen zu bearbeiten.

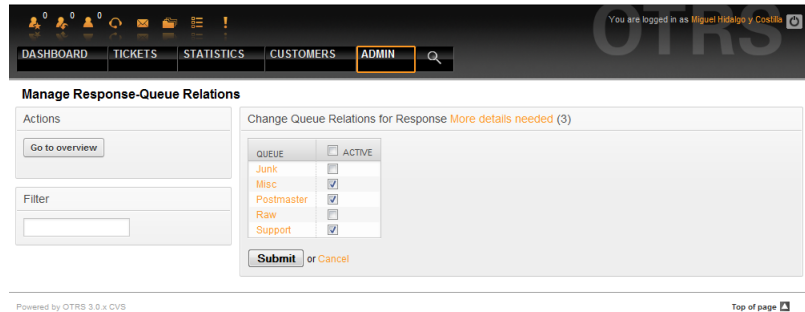


Abb.: Queue-Zuordnungen einer Antwortvorlage ändern.

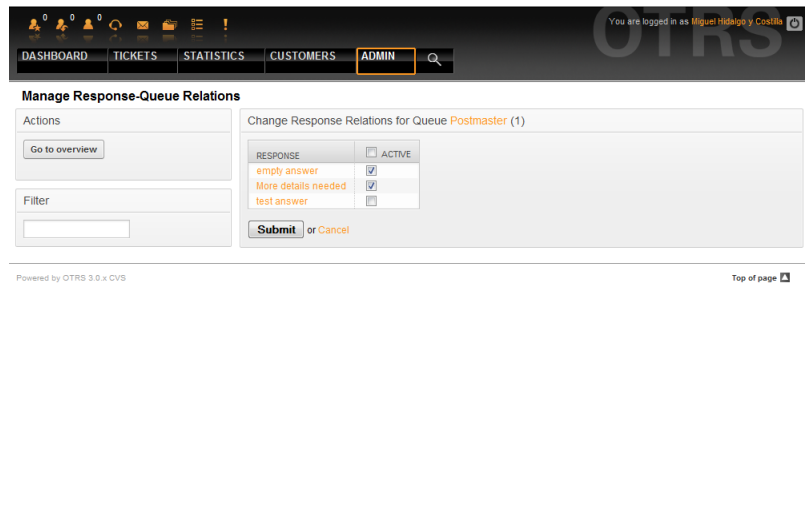


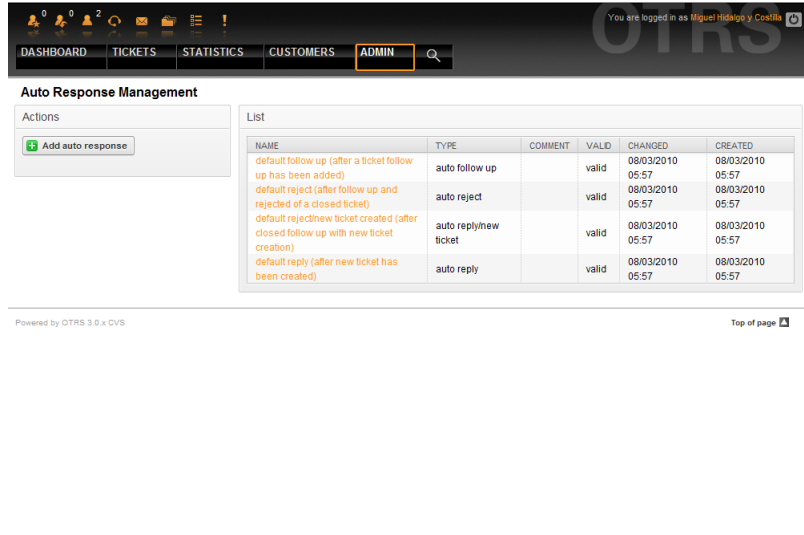
Abb.: Antwort-Zuordnungen einer Queue ändern.

Die Struktur einer Antwort ist intuitiv. Sie enthält eine mit der Queue verknüpfte Anrede, den Text der Antwort, den zitierten Ticket-Text und schließlich die Signatur der Queue.

Automatische Antworten

OTRS bietet die Möglichkeit, automatische Antworten an Kundenbenutzer zu verschicken. Automatische Antworten sind an bestimmte Ereignisse im System gebunden, z. B. an das Anlegen eines neuen Tickets in einer Queue, wenn ein Followup eines Tickets stattfindet, wenn ein Ticket geschlossen oder vom System zurückgewiesen wird. Über den Link "Auto Antworten" innerhalb des Admin-Bereiches erreichen Sie die Verwaltung der automatischen Antworten.

Der Administrationsbereich von OTRS



Auto Response Management

Actions

[Add auto response](#)

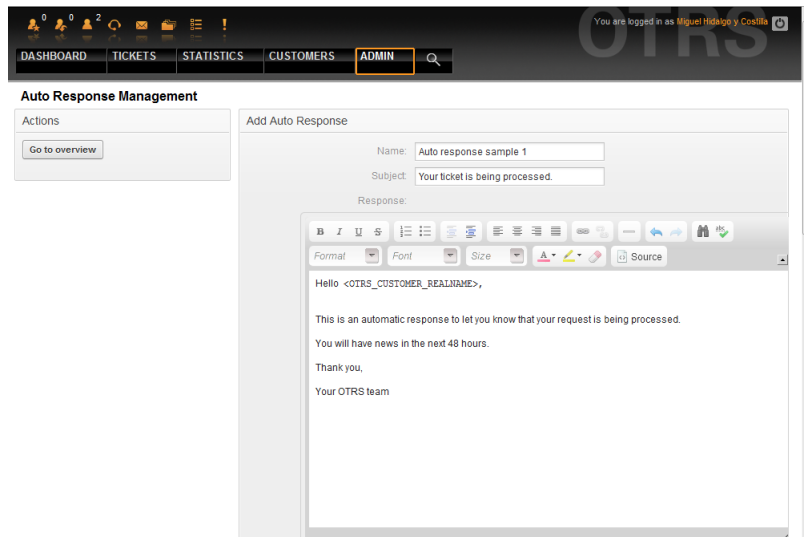
List

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 rejectnew 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

Powered by OTRS 3.0.x CVS Top of page

Abb.: Verwaltung automatischer Antworten.

Verwenden Sie den Knopf "Automatische Antwort hinzufügen", um eine neue anzulegen.



Auto Response Management

Actions

[Go to overview](#)

Add Auto Response

Name:

Subject:

Response:

Format **Font** **Size** **Source**

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

Abb.: Automatische Antwort hinzufügen.

Für die Betreffzeile und den Text von automatischen Antworten kann genauso wie bei Signaturen oder Anreden, der Inhalt mit Hilfe von OTRS-Variablen dynamisch erzeugt werden. So werden über die Variable <OTRS_CUSTOMER_EMAIL[5]> die ersten 5 Zeilen der an das System gesendeten E-Mail in die automatische Antwort eingefügt, oder durch <OTRS_CUSTOMER_FROM> die From-Zeile. Die Anmerkungen im unterem Bereich der Bildschirmmaske zur Verwaltung der automatischen Antworten listen alle OTRS-Variablen auf, die verwendet werden können.

Für jede automatische Antwort können Sie angeben, durch welches Ereignis sie ausgelöst werden soll. Folgende Ereignisse sind in einer Standardinstallation verfügbar:

Tabelle 5.3. Ereignisse für automatische Antworten

Name	Beschreibung
auto reply	Dieses Ereignis tritt ein, wenn ein neues Ticket in einer Queue angelegt wird.
auto reply/new ticket	Dieses Ereignis tritt ein, wenn ein bereits geschlossenes Ticket, z. B. durch die Antwort eines Kunden, mit einer neuen Ticketnummer erneut geöffnet wird.
auto follow up	Dieses Ereignis tritt ein, wenn ein Follow up für ein bereits vorhandenes Ticket eintrifft.
auto reject	Dieses Ereignis tritt ein, wenn ein Ticket vom System zurückgewiesen wird.
auto remove	Dieses Ereignis tritt ein, wenn ein Ticket vom System entfernt wird.

Anmerkung

In OTRS können automatische Antworten deaktiviert, aber nicht gelöscht werden. Zum Deaktivieren einer automatischen Antwort können Sie den Wert für "Gültig" auf "ungültig" oder "ungültig-temporär" setzen.

Um eine automatische Antwort einer oder mehreren Queues zuzuweisen, folgen Sie im Admin-Bereich den Link "Auto Antworten <-> Queues". Dort sind für jede Queue die verschiedenen Ereignistypen aufgelistet und es kann eine Zuordnung einer Auto-Antwort vom gleichen Ereignistyp durchgeführt bzw. entfernt werden.

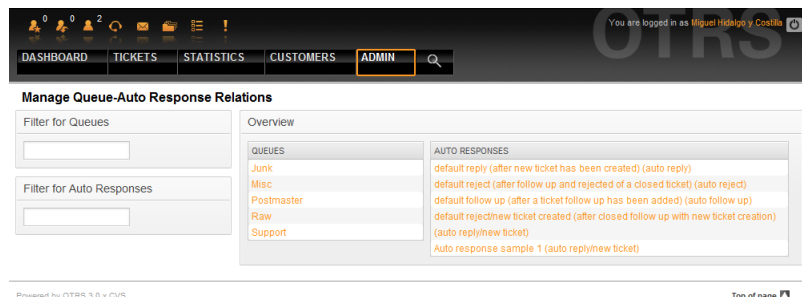


Abb.: Zuordnungen von Queues zu automatischen Antworten verwalten.

Um die Zuordnungen von automatischen Antworten zu Queues zu verändern, klicken sie auf den Namen der Queue. Durch einen Klick auf den Namen einer automatischen Antwort können Sie diese direkt bearbeiten.

Der Administrationsbereich von OTRS

The screenshot shows the OTRS Admin interface. The top navigation bar includes 'DASHBOARD', 'TICKETS', 'STATISTICS', 'CUSTOMERS', and 'ADMIN'. The 'ADMIN' tab is selected. The main content area is titled 'Manage Queue-Auto Response Relations'. On the left, there is an 'Actions' sidebar with a 'Go to overview' button. The main form is titled 'Change Auto Response Relations for Queue Junk (3)'. It contains several dropdown menus for configuring auto-response settings: 'Change "auto reply" settings' (default reply (after new ticket has be...)), 'Change "auto reply/new ticket" settings' (default reject/new ticket created (aft...)), 'Change "auto follow up" settings' (default follow up (after a ticket follow...)), 'Change "auto reject" settings' (default reject (after follow up and rej...)), and 'Change "auto remove" settings' (a dropdown menu). At the bottom of the form are 'Update' and 'Cancel' buttons. The footer indicates 'Powered by OTRS 3.0.x CVS' and 'Top of page'.

Abb.: Zuordnung von automatischen Antworten zu einer Queue bearbeiten.

E-Mail-Adressen

Um aus OTRS heraus E-Mails verschicken zu können, benötigen Sie mindestens eine gültige Mailadresse. Da in vielen Fällen eine Mailadresse nicht ausreicht, ist OTRS auch in der Lage, mit mehreren Mailadressen zu arbeiten. Eine Queue mit mehreren E-Mail-Adressen verknüpft werden, und umgekehrt. Die Adresse, die für ausgehende Nachrichten einer Queue benutzt werden soll, kann bei Erstellung oder Bearbeitung der Queue ausgewählt werden. Verwenden Sie den Link "E-Mail-Adressen" im Administrationsbereich, um die Adressen des Systems zu verwalten.

The screenshot shows the OTRS Admin interface for 'System Email Addresses Management'. The top navigation bar is the same as in the previous screenshot. The main content area is titled 'System Email Addresses Management'. On the left, there is an 'Actions' sidebar with a 'Add system address' button. Below it is a 'Note' section stating: 'All incoming email with this address in To or Cc will be dispatched to the selected queue.' The main area contains a 'List' table with the following data:

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

The footer indicates 'Powered by OTRS 3.0.x CVS' and 'Top of page'.

Abb.: Verwaltung der E-Mail-Adressen des Systems.

Im Formular zur Verwaltung der E-Mail-Adressen können Sie u.a. direkt auswählen, mit welcher Queue oder Unter-Queue die neue Mailadresse verknüpft werden soll. Diese Verknüpfung ist wichtig, da so eingehende Mails anhand der Adresse im To: Feld der entsprechenden Queue zugewiesen werden können.

Der Administrationsbereich von OTRS

System Email Addresses Management

Actions

Go to overview

Note

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

Add System Email Address

Email address: 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

Submit or Cancel

Powered by OTRS 3.0.x CVS

Top of page

Abb.: Hinzufügen einer System-E-Mail-Adresse.

Anmerkung

In OTRS können E-Mail-Adressen deaktiviert, aber nicht gelöscht werden. Sie können eine Adresse deaktivieren, indem Sie den Wert für "Gültig" auf "ungültig" oder "ungültig-temporär" setzen.

Benachrichtigungen

Kunden und Mitarbeiter können im Einstellungs-Bereich ihres Accounts festlegen, bei welchen Ereignissen Sie automatisch vom System per Mail benachrichtigt werden wollen.

Über den "Benachrichtigungen"-Link im Admin-Bereich erreichen Sie die Verwaltung der Benachrichtigungen. Mit den Filtern können Sie bestimmte Einträge finden.

Notification Management

Filter for Language

Filter for Notification

Note

Notifications are sent to an agent or a customer.

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 (Bsnrapoor)	Agent: AddNote
Bulgarian (Bsnrapoor)	Agent: Escalation
Bulgarian (Bsnrapoor)	Agent: EscalationNotifyBefore
Bulgarian (Bsnrapoor)	Agent: FollowUp
Bulgarian (Bsnrapoor)	Agent: LockTimeout
Bulgarian (Bsnrapoor)	Agent: Move
Bulgarian (Bsnrapoor)	Agent: NewTicket
Bulgarian (Bsnrapoor)	Agent: OwnerUpdate
Bulgarian (Bsnrapoor)	Agent: PendingReminder
Bulgarian (Bsnrapoor)	Agent: ResponsibleUpdate
Català	Agent: AddNote
Català	Agent: Escalation
Català	Agent: EscalationNotifyBefore
Català	Agent: FollowUp
Català	Agent: LockTimeout
Català	Agent: Move
Català	Agent: NewTicket

Abb.: Verwaltung der Benachrichtigungen.

Den Betreff und Text der Benachrichtigungen können Sie anpassen. Wählen Sie die gewünschte Benachrichtigung aus der Liste durch Klick aus. Beachten Sie: in jeder Sprache gibt es eine Benachrichtigung mit demselben Namen.

The screenshot shows the 'Edit Notification' form in the OTRS Administration interface. The form has three main sections: 'Type', 'Subject', and 'Text'. The 'Type' field is set to 'es_MX:Agent:AddNote'. The 'Subject' field contains the text 'New note (<OTRS_CUSTOMER_SUBJECT24>)'. The 'Text' field contains a template with OTRS variables: '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>', and 'Your OTRS Notification Master'. The interface also shows a sidebar with 'Notification Management' and 'Actions'.

Abb.: Bearbeiten einer Benachrichtigung.

Auch innerhalb der Benachrichtigungen können die Textinhalte mit Hilfe der OTRS-Variablen dynamisch aufbereitet werden. Innerhalb der Anmerkungen im unteren Bereich der Bildschirmmaske zur Benachrichtigungs-Verwaltung werden die verschiedenen zur Verfügung stehenden Variablen und deren Verwendung aufgelistet und näher erklärt.

Ebenso ist es möglich, Benachrichtigungen zu erstellen, die auf Ereignissen basieren. Hier können Sie detailliert einstellen, wann diese Benachrichtigung geschickt werden soll. Zudem kann ein weites Spektrum von Parametern konfiguriert werden, wie: Empfängergruppen(n), Benutzer, Rollen, E-Mail-Adressen; Ereignistyp, der die Benachrichtigung auslöst, Ticket-Typ, Status, Priorität, Sperre, Service, SLA, usw.

Um eine Übersicht aller ereignisbasierten Benachrichtigungen zu erhalten, klicken Sie auf "Benachrichtigungen (Ereignis)" im Administrationsbereich.

The screenshot shows the 'List' view of notifications in the OTRS Administration interface. The table lists notifications with columns: NAME, COMMENT, VALID, CHANGED, CREATED, and DELETE. Two notifications are listed: 'My test notification' and 'My test notification 2'.

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	

Abb.: Ereignisbasierte Benachrichtigungen verwalten.

Erstellen Sie eine neue Benachrichtigung durch Klick auf den "Hinzufügen"-Knopf.

The screenshot shows the OTRS Admin interface with the 'ADMIN' tab selected. The 'Notification Management' section is active, displaying the 'Add Notification' form. The form includes fields for 'Name', 'Recipient groups' (with a dropdown menu showing options like 'Agent (All with write permissions)', 'Agent (Owner)', 'Agent (Responsible)', and 'Customer'), 'Recipient agents' (with a list of users including 'Hidalgo y Costilla Miguel (miguel.hidalgo)' and 'OTRS Admin (root@localhost)'), 'Recipient groups' (with a list of groups like 'admin' and 'fill-up-group1004052'), 'Recipient roles' (with a dropdown menu), 'Recipient email addresses' (with a text input field), and 'Event' (with a dropdown menu showing options like 'ArticleBounce', 'ArticleCreate', 'ArticleFreeTextUpdate', and 'ArticleSend').

Abb.: Ereignisbasierte Benachrichtigung einrichten.

Auch in den ereignisbasierten Benachrichtigungen können Sie mit den speziellen Variablen dynamische Texte erzeugen. Die Referenz der Variablen sehen Sie im unteren Bildschirmbereich.

S/MIME

Mit OTRS ist es möglich, Tickets mit Hilfe von SMIME zu ent- und verschlüsseln bzw. Tickets zu signieren. Bevor SMIME allerdings systemweit genutzt werden kann, muss das Feature mit Hilfe einiger Konfigurationsparameter aktiviert und konfiguriert werden.

Über den "SMIME"-Link im Admin-Bereich von OTRS erreichen Sie die Verwaltung der SMIME-Zertifikate. Es können Zertifikate und private Fingerprints hinzugefügt und entfernt werden und eine Suche in den Zertifikaten ist möglich.

The screenshot shows the OTRS Admin interface with the 'ADMIN' tab selected. The 'S/MIME Management' section is active. It features an 'Actions' panel on the left with a search bar and buttons for 'Add private key' and 'Add certificate'. Below this is a 'Note' section with text explaining that users can directly edit the certification and private keys in the file system, with a link to the Wikipedia page on S/MIME. On the right, the 'Results' panel shows a table with columns 'TYPE', 'SUBJECT', 'HASH/FINGERPRINT', 'CREATE/EXPIRES', and 'DELETE'. The table currently displays 'No data found.' At the bottom of the page, there is a footer indicating 'Powered by OTRS 3.0.x CVS' and a 'Top of page' link.

Abb.: S/MIME-Verwaltung.

PGP

Mit OTRS ist es möglich, Tickets mit Hilfe von PGP zu ent- und verschlüsseln bzw. zu signieren. Bevor PGP allerdings systemweit genutzt werden kann, muss das Feature mit Hilfe einiger Konfigurationsparameter aktiviert und konfiguriert werden.

Über den "PGP"-Link im Admin-Bereich von OTRS erreichen Sie die Verwaltung des Schlüsselrings Ihres Systems. Es können Schlüssel und Signaturen hinzugefügt und entfernt werden und eine Suche innerhalb des Schlüsselrings ist möglich.

The screenshot shows the OTRS Admin interface with the 'ADMIN' tab selected. The 'PGP Management' section is active, displaying a 'Result' table with columns: TYPE, IDENTIFIER, BIT, KEY, FINGERPRINT, CREATED, EXPIRES, and DELETE. The table is currently empty, showing 'No data found.' Below the table, there is a 'Note' section with a warning about editing the keyring configuration in SysConfig and a link to the Wikipedia page on Pretty Good Privacy (PGP).

Abb.: PGP-Verwaltung.

Status

Über den Link "Status" können Sie die verschiedenen Ticket-Status bearbeiten, die Sie in OTRS verwenden möchten.

The screenshot shows the OTRS Admin interface with the 'ADMIN' tab selected. The 'State Management' section is active, displaying a 'List' table with columns: NAME, TYPE, COMMENT, VALID, CHANGED, and CREATED. The table lists various ticket states and their configurations.

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

Abb.: Status-Verwaltung.

In einer Standardinstallation sind folgende Status definiert:

- erfolgreich geschlossen
- erfolglos geschlossen
- zusammengefasst
- neu
- öffnen
- warten auf erfolgreich schließen
- warten auf erfolglos schließen
- warten zur Erinnerung
- entfernt

Jeder Status ist mit einem Typ verküpft. Auch bei dem Neuanlegen eines Status muss ein Typ angegeben werden. Die standardmäßig verfügbaren Status-Typen sind:

- geschlossen
- zusammengefasst
- neu
- öffnen
- warten auto
- warten zur Erinnerung
- entfernt

Die SysConfig (Verwaltung der Systemkonfiguration)

In der SysConfig werden die meisten Konfigurationseinstellungen von OTRS vorgenommen.

Über den "SysConfig"-Link im Admin-Bereich von OTRS erreichen sie die grafische Administrationsoberfläche. Über diese können Sie eigene Konfigurationsdateien in das System integrieren bzw. Ihre persönlichen Änderungen an der Standardkonfiguration in eine Datei sichern. Nahezu alle Konfigurationsparameter des OTRS Frameworks und der zusätzlich installierten Module können eingesehen und geändert werden, die Navigation durch die Vielzahl der Konfigurationsparameter wurde durch die Einteilung in Module und Gruppen übersichtlich aufgeteilt. Weiterhin ist es möglich, eine Suche über alle Konfigurationsparameter hinweg durchzuführen, so dass einzelne Parameter schnell aufgefunden und bearbeitet werden können.

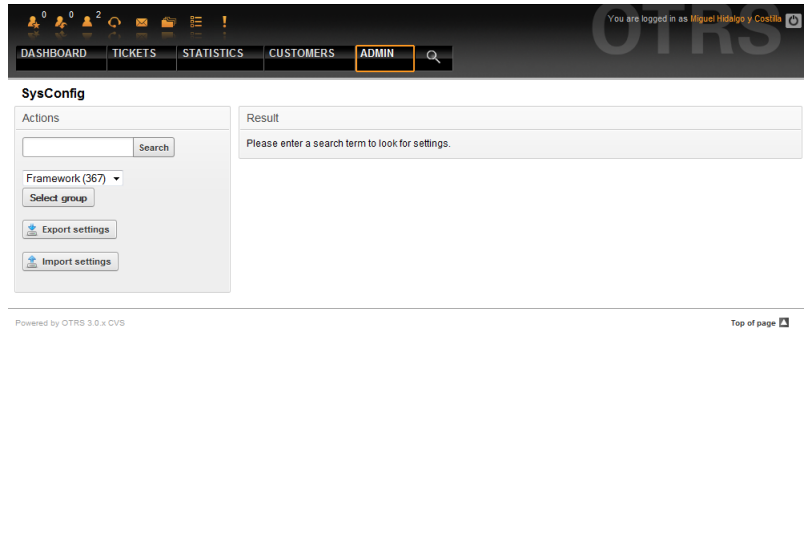


Abb.: Das grafische Konfigurationsfrontend von OTRS (SysConfig).

Das Kapitel Nähere Beschreibung der grafischen Administrationsoberfläche geht ausführlicher auf das grafische Konfigurations-Frontend ein.

Einrichten von Mail-Konten

Es gibt mehrere Möglichkeiten, Mails in OTRS einzuspeisen. Neben der Nutzung des PostMaster.pl-Skripts, welches die Mails direkt in das System piped, können auch Mail-Konten abgefragt werden, an die Mails für Ihr Ticket-System gehen. Über den Link "PostMaster Mail Account" innerhalb des Admin-Bereiches erreichen Sie die Verwaltung der Mail-Konten, die OTRS automatisch auf neue Nachrichten überprüfen soll. OTRS unterstützt dabei die Mail-Protokolle POP3, POP3S, IMAP und IMAPS.

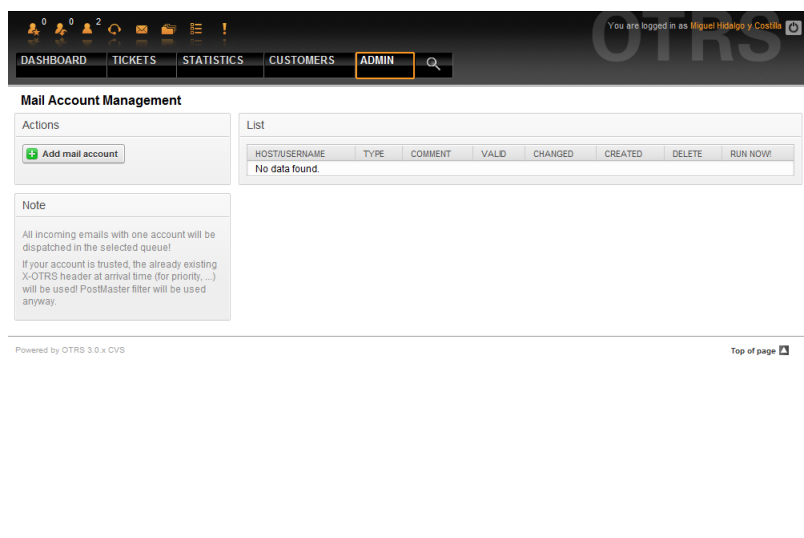


Abb.: Verwaltung von Mail-Konten.

Im Kapitel über die Mailkonfiguration für OTRS wird die Einrichtung der Mail-Konten näher erläutert.

Einkommende Nachrichten filtern

Damit Sie automatisch Mails vorsortieren können bzw. ungewünschte Mails gar nicht in die Queues des Ticket-Systems gelangen, können Filterregeln für eingehende Nachrichten festgelegt werden. Dabei spielt es keine Rolle, ob die Mails über Mail-Konten oder mit Hilfe des `PostMaster.pl` Skriptes ins System gelangen. Der Link "PostMaster Filter" innerhalb des Admin-Bereiches von OTRS ruft die Verwaltung für die Filterregeln auf.

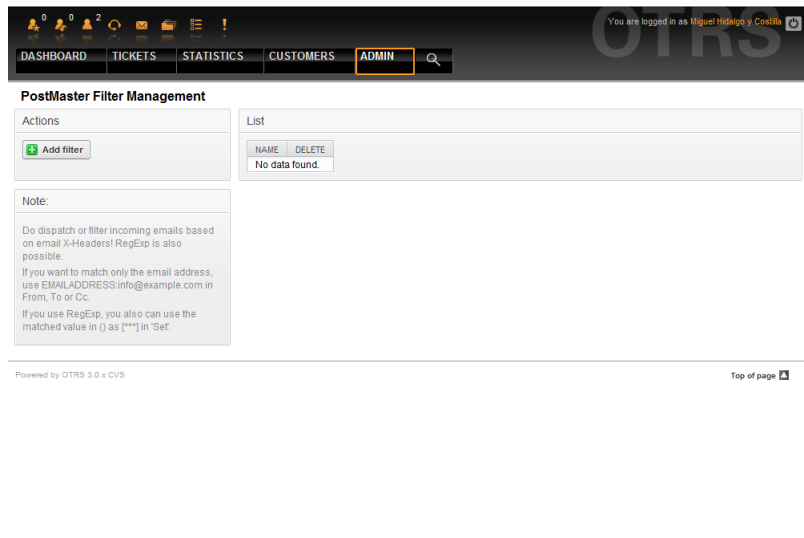


Abb 5.49. Verwaltung von Filterregeln für eingehende E-Mails.

Eine Filterregel besteht aus einem oder mehreren Kriterien die erfüllt werden müssen, damit die Filterregel greift, und aus einer oder mehreren Aktionen die ausgeführt werden, nachdem die Filterkriterien erfüllt wurden. Sie können für die Kopfzeilen und den Body einer Nachricht Filterregeln festlegen, also z. B. nach bestimmten Headern suchen oder nach Zeichenketten bzw. regulären Ausdrücken innerhalb der Mail. Alle Aktionen für eine Filterregel werden über sog. X-OTRS-Header gesteuert, die in die Mail eingefügt werden. Das Ticket-System wertet die X-OTRS-Header aus und nimmt die entsprechenden Aktionen vor. Mit den X-OTRS-Headern kann z. B. die Priorität einer Nachricht geändert werden, die Mail in eine bestimmte Queue einsortiert oder die Mail komplett verworfen werden, usw. Die folgende Tabelle listet die verschiedenen X-OTRS-Header, deren zulässige Werte und die Bedeutung der einzelnen Header auf.

Hinweis: Sie können auch X-OTRS-FollowUp-* header für Follow-Up-E-Mails verwenden.

Tabelle 5.4. Funktion der verschiedenen X-OTRS-Header

Name	Mögliche Werte	Beschreibung
X-OTRS-Priority:	1 very low, 2 low, 3 normal, 4 high, 5 very high	Legt die Priorität eines Tickets fest.
X-OTRS-Queue:	Name einer Queue des Systems.	Legt die Queue fest, in die das Ticket einsortiert werden soll. Wird mit dem X-OTRS-Queue-Header eine Queue voreingestellt, so hat diese Einstellung Vorrang vor allen anderen Filterregeln, die sich auf Queues beziehen.

Der Administrationsbereich
von OTRS

Name	Mögliche Werte	Beschreibung
X-OTRS-Lock:	lock, unlock	Setzt den "gesperrt"-Status eines Tickets.
X-OTRS-Ignore:	Yes oder True	Wird dieser Header gesetzt, wird die Mail komplett ignoriert und gelangt somit nie als Ticket in das System.
X-OTRS-State:	new, open, closed successful, closed unsuccessful, ...	Setzt den nächsten Status eines Tickets.
X-OTRS-State-PendingTime:	Z. B. 2010-11-20 00:00:00	Setzt die Wartezeit eines Tickets (Sie sollten dann ebenfalls einen "Warten"-Status per X-OTRS-State setzen).
X-OTRS-Type:	default (systemabhängig)	Setzt den Typ eines Tickets (wenn Ticket::Type aktiviert ist).
X-OTRS-Service:	(systemabhängig)	Setzt den Service eines Tickets (wenn Ticket::Service aktiv ist).
X-OTRS-SLA:	(systemabhängig)	Setzt das Service-Level-Abkommen eines Tickets (wenn Ticket::Service aktiv ist).
X-OTRS-CustomerUser:	CustomerUser	Legt den Kunden-Benutzer fest, dem das Ticket zugeordnet werden soll.
X-OTRS-CustomerNo:	CustomerNo	Legt die Kunden-ID fest, die dem Ticket zugeordnet werden soll.
X-OTRS-SenderType:	agent, system, customer	Legt die Art des Ticket-Absenders fest.
X-OTRS-ArticleType:	email-external, email-internal, email-notification-ext, email-notification-int, phone, fax, sms, webrequest, note-internal, note-external, note-report	Legt den Typ des Artikles für das eingehende Ticket fest.
X-OTRS-DynamicField-<DynamicFieldName>:	Depends on Dynamic Field configuration (Text: Notebook, Date: 2010-11-20 00:00:00, Integer: 1)	Saves an additional info value for the ticket on <DynamicFieldName> Dynamic Field.
X-OTRS-Loop:	True	Ist dieser X-OTRS-Header gesetzt, wird keine automatische Antwort an den Absender des neuen Tickets geschickt, z. B. um Mailschleifen zu vermeiden.

Sie müssen für jede Regel einen Namen vergeben. Darunter legen Sie in der Sektion für "Treffer" die verschiedenen Kriterien fest, nach denen gefiltert werden soll. Wählen Sie aus den Listboxen für "Header 1", "Header 2" usw. aus, an welcher Stelle in der eingehenden Nachricht Sie suchen möchten und tragen Sie rechts neben dem entsprechenden Header den Wert ein, nach dem gesucht werden soll. In der Sektion "Setzen" legen Sie die Aktionen fest, die ausgeführt werden sollen, wenn die Filterkriterien zutreffen. Hier können Sie für "Header1", "Header 2" usw. den entsprechenden X-OTRS-Header auswählen und rechts daneben den entsprechenden Wert eintragen.

The screenshot shows the OTRS Admin interface. At the top, there's a navigation bar with 'DASHBOARD', 'TICKETS', 'STATISTICS', 'CUSTOMERS', and 'ADMIN'. The 'ADMIN' tab is selected. Below the navigation bar, the 'PostMaster Filter Management' section is active. On the left, there's a 'Note' box with instructions on how to use the filter. The main area is titled 'Add PostMaster Filter'. It contains a 'Filtername' field with the value 'My sample filter', a 'Stop after match' dropdown set to 'No', and a 'Filter Condition' section. The 'Filter Condition' section has four headers: Header 1 (From: *@independence.com), Header 2, Header 3, and Header 4. Below the 'Filter Condition' section is the 'Set Email Headers' section, which also has four headers: Header 1 (X-OTRS-Queue: Special queue), Header 2, Header 3, and Header 4.

Abb.: Filter-Regeln für eingehende Nachrichten anlegen.

Beispiel 5.1. Aussortierung von Spammails in eine bestimmte Queue

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

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

Ausführen von automatisierten Jobs mit Hilfe des GenericAgents

Der GenericAgent ist ein Tool zum automatischen Ausführen von Aufgaben, die normalerweise ein richtiger Agent manuell durchführen müsste. Er kann z. B. bestimmte Tickets in einer Queue schließen oder Benachrichtigungen für eskalierte Tickets versenden.

Um zum GenericAgent zu gelangen, verwenden Sie die Verknüpfung "GenericAgent" in der Administrationsoberfläche von OTRS. Eine Liste verfügbarer Jobs wird angezeigt, welche bearbeitet, ausgeführt oder gelöscht werden können.

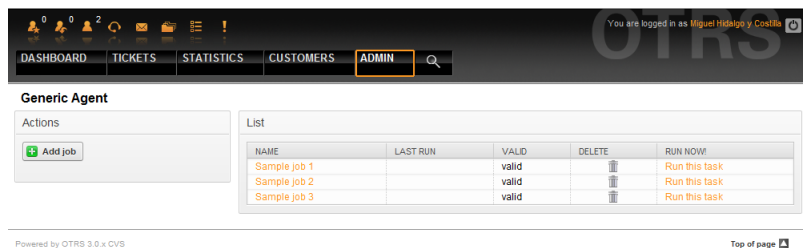


Abb.: Jobliste des GenericAgent.

Über die Bildschirmmaske zum Erstellen eines neuen Jobs kann der Zeitplan für die Ausführung des Jobs eingestellt werden. Desweiteren kann über verschiedene Kriterien festgelegt werden, welche Tickets vom Job erfasst werden sollen. Schließlich ist es möglich, die neuen Eigenschaften der vom Job betroffenen Tickets einzustellen.

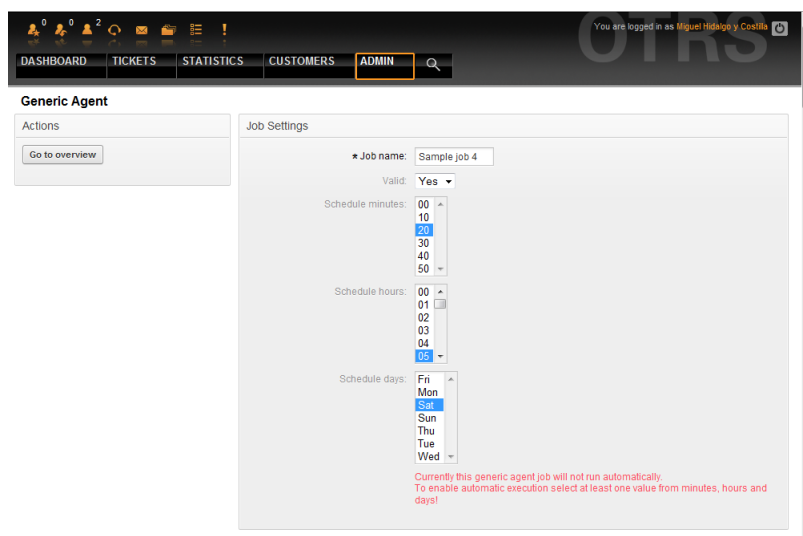


Abb.: Anlegen eines GenericAgent-Jobs.

Nachdem der Job gespeichert wurde, werden alle Tickets aufgeführt, die durch den Job verändert werden. Diese Auflistung dient zur Übersicht, ob der Job richtig funktioniert, es werden noch keine Veränderungen vorgenommen. Erst nach der Übernahme des Jobs in die Job-Liste, wird der Job aktiviert.

Administrator-E-Mail

Um als OTRS-Administrator eine Mail an bestimmte Benutzer (Agenten) oder Benutzergruppen im System zu versenden, folgen Sie dem Link "Admin Notification".

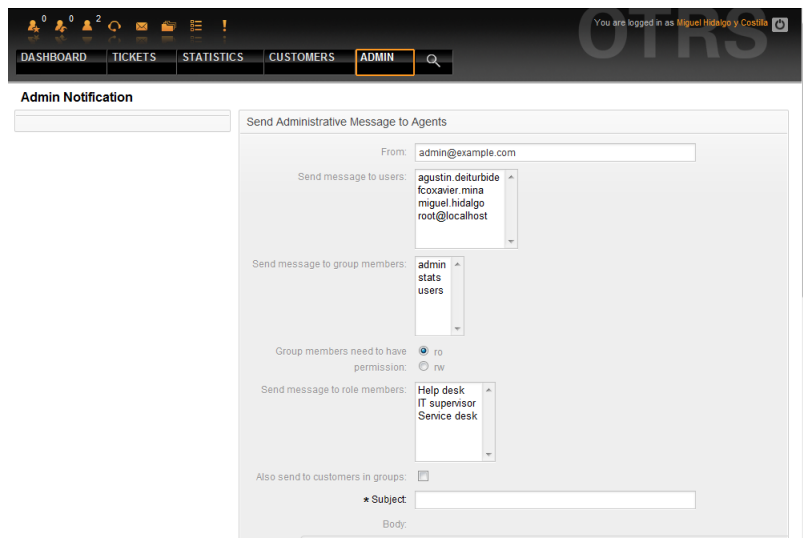
The screenshot shows the OTRS web interface. At the top, there is a navigation bar with tabs: DASHBOARD, TICKETS, STATISTICS, CUSTOMERS, and ADMIN (which is highlighted). Below the navigation bar, the 'Admin Notification' section is active. It contains a form titled 'Send Administrative Message to Agents'. The form has several fields: 'From:' with the value 'admin@example.com'; 'Send message to users:' with a dropdown menu showing 'agustin.deiturbide', 'fcoaxavier.mina', 'miguel.hidalgo', and 'root@localhost'; 'Send message to group members:' with a dropdown menu showing 'admin', 'stats', and 'users'; 'Group members need to have permission:' with radio buttons for 'rw' (selected) and 'nw'; 'Send message to role members:' with a dropdown menu showing 'Help desk', 'IT supervisor', and 'Service desk'; and 'Also send to customers in groups:' with a checkbox. At the bottom of the form, there are fields for '* Subject:' and 'Body:'.

Abb.: Administrator-Benachrichtigung.

Es wird ein Formular geladen, in das Sie die Absenderadresse, den Betreff und den Inhalt der Administratoren-Mitteilung eintragen können. Weiterhin können Sie aus der Tabelle auswählen, an welche Benutzer und / oder Benutzergruppen bzw. Rollen die Mitteilung gesendet werden soll.

Sitzungsverwaltung

Um eine Übersicht über die gerade im System angemeldeten Benutzer und deren Sitzungseigenschaften zu erhalten, folgen Sie dem Link "Sitzungsverwaltung" der Administrationsoberfläche.

Der Administrationsbereich von OTRS

The screenshot shows the OTRS Admin interface. At the top, there's a navigation bar with tabs: DASHBOARD, TICKETS, STATISTICS, CUSTOMERS, and ADMIN (selected). Below the navigation bar, the 'Session Management' section is active. It features a summary on the left and a list of sessions on the right.

Session Management Summary:

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

Session List:

SESSION	TYPE	USER	KILL
105b77fe2de2a3d4f78ed8780bc6817717	Agent	Miguel Hidalgo y Costilla	Kill this session
10848c1d962d1ef8b5e9de44f85d90a7	Customer	Leona Vicario	Kill this session
10e5cd5fb565e19ac075f59e99b5941	Customer	Ignacio López Rayón	Kill this session

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

Abb.: Sitzungsverwaltung.

Innerhalb der Sitzungsverwaltung werden allgemeine Informationen zu allen Sitzungen im System ausgegeben, also z. B. wie viele Sitzungen gerade insgesamt aktiv sind oder wie viele Agenten- und Kunden-Sitzungen laufen, usw. Es besteht die Möglichkeit mit Hilfe des "Alle Sitzungen löschen"-Schalters die Sitzungen aller angemeldeten Benutzer zu beenden. Weiterhin können detaillierte Informationen für jede einzelne Sitzung abgerufen und einzelne Sitzungen gelöscht werden.

The screenshot shows the OTRS Admin interface with the 'Session Management' section. The 'Detail View for SessionID : 105b77fe2de2a3d4f78ed8780bc6817717 - Miguel Hidalgo y Costilla' is displayed.

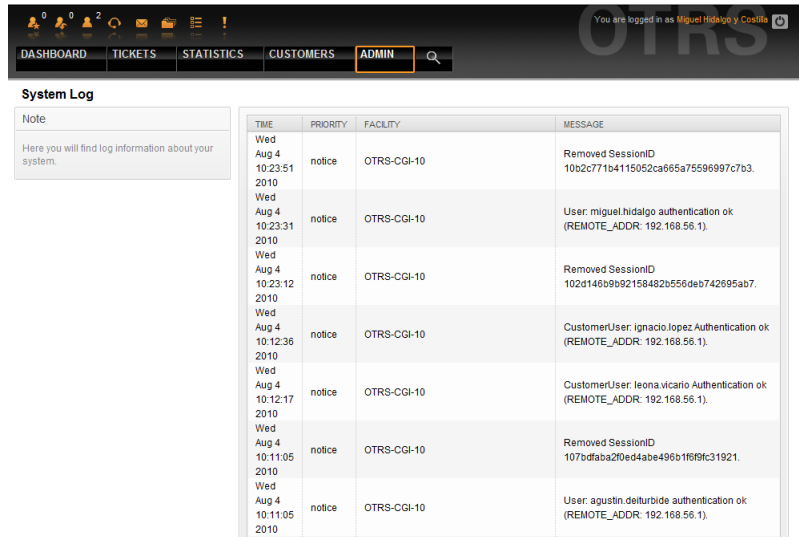
Session Details:

KEY	VALUE
ChangeTime	2010-08-03 06:01:50
CreateTime	2010-08-03 06:01:50
OutOfficeEndDay	4
OutOfficeEndMonth	8
OutOfficeEndYear	2010
OutOfficeStartDay	3
OutOfficeStartMonth	8
OutOfficeStartYear	2010
SessionID	105b77fe2de2a3d4f78ed8780bc6817717
UserChallengeToken	44305769205c0299e0c213b15072eb6
UserEmail	miguel.hidalgo@freedom.com
UserFirstname	Miguel
UserID	2
UsersGroupRo[admin]	Yes
UsersGroupRo[stats]	Yes
UsersGroupRo[users]	Yes
UsersGroup[admin]	Yes
UsersGroup[stats]	Yes
UsersGroup[users]	Yes
UserLanguage	en
UserLastLogin	1280935411
UserLastLoginTimestamp	2010-08-04 10:23:31
UserLastPw	xxxxxxx
UserLastRequest	1280935483
UserLastname	Hidalgo y Costilla
UserLogin	miguel.hidalgo
UserPw	xxxxxxx

Abb.: Sitzungsdetails.

System Log

Der Link "System Log" ermöglicht es, die letzten Logeinträge des Ticket-Systems über die Weboberfläche einzusehen.



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 107bdfaba20ed4abe496b1f6f9fc31921.
Wed Aug 4 10:11:05 2010	notice	OTRS-CGI-10	User: agustin.delturbide authentication ok (REMOTE_ADDR: 192.168.56.1).

Abb.: System Log.

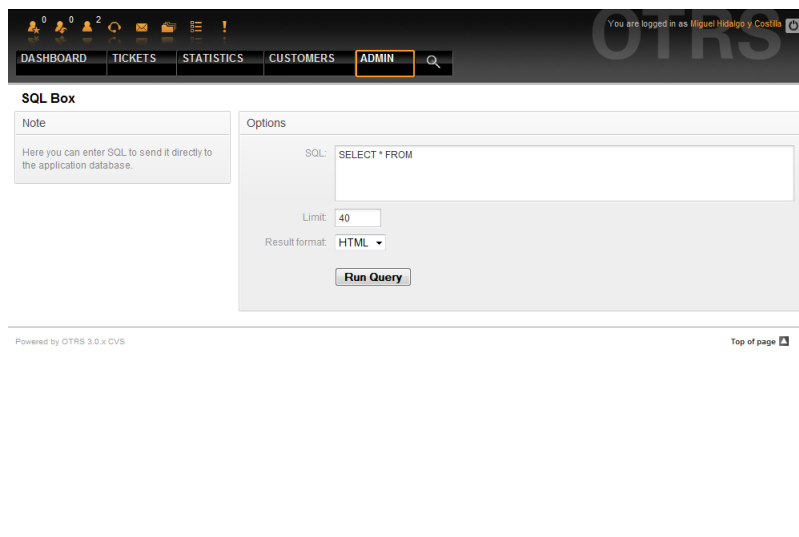
Ein Logeintrag setzt sich aus der Zeit, der Priorität, der betroffenen Systemkomponente und der eigentlichen Meldung zusammen.

Anmerkung

Die System Logs können nur auf Unix- oder Linux-Systemen über das Web-Interface eingesehen werden, unter Windows-Betriebssystemen können Sie stattdessen die Datei `[install_dir]otrs\var\log\otrs.log` mit einem Text-Editor öffnen.

SQL-Abfragen mit Hilfe der SQL Box

Über den Link "SQL Box" kann eine Bildschirmmaske aufgerufen werden, die es ermöglicht direkt mit SQL-Statements den Inhalt von Tabellen der OTRS-Datenbank abzurufen. Es sind nur SELECT-Abfragen möglich, d.h. die Tabellen können auf diesem Weg nicht verändert werden.



SQL Box

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

Options

SQL:

Limit:

Result format:

Powered by OTRS 3.0.x CVS Top of page

Abb.: SQL Box.

Paketverwaltung

Mit der "Paketverwaltung" in der Administrationsoberfläche können Sie Pakete installieren und verwalten, die die Funktionalität von OTRS erweitern. Eine Liste von verfügbaren Erweiterungsmodulen von OTRS finden Sie unter Zusätzliche Applikationen.

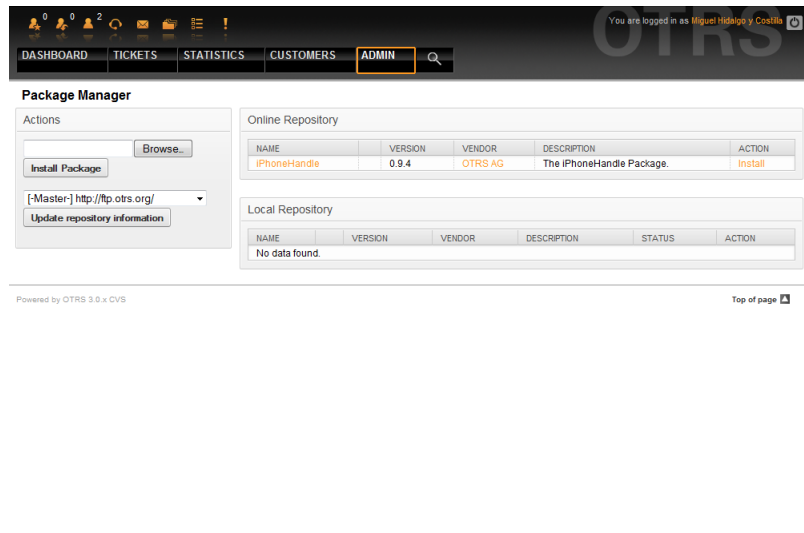


Abb.: Paketverwaltung.

In der Paketverwaltung können Sie die bereits installierten Pakete mit ihrer Versionsinformation einsehen.

Sie können Pakete von einem entfernten Server installieren, indem Sie das Paketverzeichnis im Abschnitt *Online-Paketverzeichnis* auswählen und auf *Verzeichnisinformation aktualisieren* klicken. Die verfügbaren Pakete werden rechts tabellarisch dargestellt. Um eines zu installieren, klicken Sie auf "Installieren". Nach der Installation wird das Paket im Abschnitt *Lokales Verzeichnis* dargestellt.

Um ein installiertes Paket zu aktualisieren, können Sie die Funktion "Aktualisieren" benutzen, die bei installierten Paketen angezeigt wird, von denen eine neuere Version verfügbar ist. Klicken Sie einfach auf "Aktualisieren", und die neue Version wird in Ihrem System installiert.

In manchen Fällen, wenn beispielsweise ein OTRS-System nicht mit dem Internet verbunden ist, können Sie auch Pakete installieren, die lokal heruntergeladen wurden. Klicken Sie auf den *Datei auswählen*-Knopf in der "Aktionen"-Liste, und wählen Sie die .opm-Datei auf Ihrem Speichermedium aus. Verwenden Sie dann *Paket installieren*. Nach der Installation wird das Paket im Abschnitt *Lokales Verzeichnis* angezeigt. Auf dieselbe Weise können Sie ein bereits installiertes Paket aktualisieren.

In besonderen Fällen ist es erforderlich, die Paketverwaltung zu konfigurieren, zum Beispiel für die Verwendung eines Proxy-Servers, oder um auf ein lokales Verzeichnis zuzugreifen. Verwenden Sie dazu die Einstellungen unter Framework:Core::Package.

Web-Services

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

Figure: The graphical interface for web services.

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

Dynamic Fields

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

Figure: The dynamic fields overview screen with some dynamic fields.

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.

Anmerkung

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.

Kapitel 6. Systemkonfiguration

Die Konfigurationsdateien von OTRS

Alle Konfigurationsdateien des OTRS-Frameworks befinden sich innerhalb des Verzeichnisses `Kernel` bzw. in Unterverzeichnissen dieses Directorys. Bis auf die Datei `Kernel/Config.pm` sollten Sie keine Konfigurationsdatei manuell verändern, da alle anderen Dateien beim Updaten des Systems überschrieben werden und so Ihre eigenen Einstellungen verloren gehen. Übertragen Sie lediglich die Parameter aus den anderen Dateien nach `Kernel/Config.pm` und passen Sie die Parameter Ihren Wünschen entsprechend an.

Die Datei `Kernel/Config/Defaults.pm` enthält die Konfigurationsparameter für den OTRS-Framework. In ihr finden Sie grundlegende Einstellungen wie die Mailkonfiguration, die Datenbankbindung, Standardsprache o.ä. In der Datei `Kernel/Config/Files/Ticket.pm` sind alle Konfigurationsparameter für das Ticketsystem aufgeführt.

Das Verzeichnis `Kernel/Config/Files` enthält weitere Konfigurationsdateien, die beim Starten von OTRS eingelesen werden. Sind zusätzliche Module wie der Filemanager oder der Webmailer installiert, liegen die Konfigurationsdateien dieser Applikationen ebenfalls in `Kernel/Config/Files`.

Wird die Web-Oberfläche von OTRS aufgerufen, werden die xml-Dateien in `Kernel/Config/Files` in alphabetischer Reihenfolge ausgelesen und die Einstellungen des Frameworks und der evtl. zusätzlich installierten Applikationen geladen. Anschließend werden die Einstellungen in den Dateien `Kernel/Config/Files/ZZZAAuto.pm` und `Kernel/Config/Files/ZZZAuto.pm` ausgewertet. Beide Dateien werden vom grafischen Konfigurations-Frontend angelegt und sollten auf keinem Fall manuell geändert werden. Zuletzt wird die Datei `Kernel/Config.pm` mit den von Ihnen individuell angepassten Konfigurationsparametern ausgewertet, so dass auf jeden Fall Ihre eigenen Einstellungen geladen werden.

Konfiguration des Systems mit Hilfe des grafischen Konfigurations-Frontends

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).

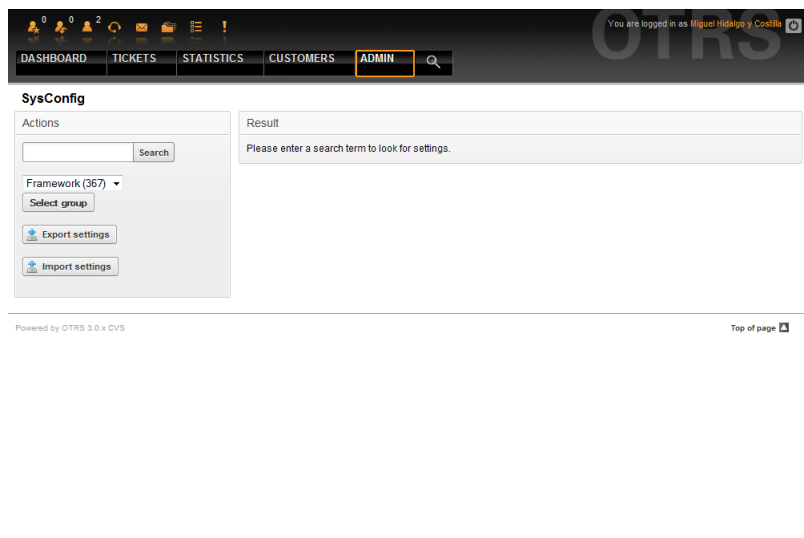


Abb.: Die grafische System-Konfigurationsoberfläche.

Da OTRS mittlerweile über mehr als 600 verschiedene Konfigurationsparameter verfügt, bietet das Konfigurations-Frontend mehrere Möglichkeiten zur schnellen Auffindung der gewünschten Einstellung. Es kann über alle Konfigurationsparameter hinweg nach einem bestimmten Stichwort gesucht werden. Bei der Suche werden neben dem Namen des Konfigurationsparameters auch die Beschreibungen ausgewertet, eine Einstellung kann also auch gefunden werden, wenn ihr Name nicht bekannt ist.

Weiterhin wurden die verschiedenen Konfigurationsparameter in Haupt- und Untergruppen unterteilt. Die Hauptgruppe stellt die Applikation dar, für die der Konfigurationsparameter zuständig ist, also z. B. "Framework" für das OTRS-Framework oder "Ticket" für das Ticketsystem. Die Untergruppen einer Hauptgruppe können eingesehen werden, indem die Gruppe bzw. Applikation aus der dazu vorgesehenen Listbox ausgewählt und der "Zeigen"-Knopf gedrückt wird.

Für jeden einzelnen Konfigurationsparameter kann über eine Checkbox festgelegt werden, ob er vom System beachtet werden soll oder nicht. Wird eine Einstellung verändert, kann die Änderung mit Hilfe des "Aktualisieren"-Buttons übernommen werden. Eine Einstellung kann mit Hilfe des "Rücksetzen"-Schalters auf ihren Default-Wert zurückgesetzt werden.

Für die Sicherung aller von Ihnen vorgenommenen Änderungen, kann eine .pm-Datei heruntergeladen werden, die alle vom Standard abweichenden Konfigurationsparameter Ihres Systems enthält. Dieselbe Datei können Sie ebenfalls über die Konfigurationsoberfläche eines frisch installierten Systems zurück spielen und so alle Einstellungen wiederherstellen.

Anmerkung

Die Einstellungen für die Datenbankankbindung können aus Sicherheitsgründen nicht über das grafische Konfigurations-Frontend geändert werden und müssen manuell in die Datei `Kernel/Config.pm` eingefügt werden.

Kapitel 7. E-Mails versenden/ empfangen

E-Mails versenden

Via Sendmail (Standard)

OTRS ist in der Lage, E-Mails via Sendmail (z. B. Sendmail [<http://www.sendmail.org/>], Postfix [<http://www.postfix.org/>], Qmail [<http://www.qmail.org>] oder Exim [<http://www.exim.org>]) zu versenden. Die Standard-Konfiguration sollte gleich ohne Probleme funktionieren.

Die Konfiguration kann über die grafische Administrationsoberfläche (Framework::Core::Sendmail) vorgenommen werden

Via SMTP server/smarthost

Wenn kein sendmail-Binary zur Verfügung steht, kann OTRS E-Mails via SMTP (Simple Mail Transfer Protocol / RFC 821 [<http://www.ietf.org/rfc/rfc821.txt>]) oder Secure SMTP versenden. Diese Möglichkeit kann hauptsächlich auf Nicht-Unix-Plattformen (z. B. Win32) genutzt werden.

Die SMTP-Einstellungen können über die SysConfig vorgenommen werden (Framework::Core::Sendmail). Wenn SMTPS nicht als Option auftaucht, fehlen benötigte Perl-Module. Lesen Sie in diesem Fall im Abschnitt "Installation benötigter Perl-Module" nach.

E-Mails empfangen

E-Mail-Konten, die über die OTRS-Oberfläche konfiguriert werden

OTRS ist in der Lage, E-Mails von POP3, POP3S, IMAP und IMAPS E-Mail-Konten zu empfangen.

Konfigurieren Sie Ihre Mail-Konten im Admin-Bereich von OTRS im Bereich "PostMaster Mail Account".

Beim Anlegen eines neuen Mail-Accounts muss der Mail-Server, ein Login und ein Kennwort angegeben werden. Zusätzlich muss der Mail-Server-Typ angegeben werden, es werden die Protokolle POP3, POP3S, IMAP und IMAPS unterstützt. Wenn Sie den gewünschten Typ nicht als Option zur Verfügung haben, fehlen benötigte Perl-Module. Lesen Sie in diesem Fall im Abschnitt "Installation benötigter Perl-Module" nach.

The screenshot shows the OTRS Admin interface. At the top, there's a navigation bar with 'DASHBOARD', 'TICKETS', 'STATISTICS', 'CUSTOMERS', and 'ADMIN'. The 'ADMIN' tab is selected. Below the navigation bar, there's a 'Mail Account Management' section. On the left, there's a 'Note' box with text about incoming emails and X-OTRS headers. The main area is titled 'Add Mail Account' and contains a form with the following fields:

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

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

Abb 7.1. Einen E-Mail-Account hinzufügen.

Wählen Sie für "Vertraut" den Wert "Ja" aus, dann werden die sog. X-OTRS-Header-Einträge ausgewertet und angewendet, sofern derartige Header-Einträge in einer abgerufenen Nachricht vorhanden sind. Da mit Hilfe der X-OTRS-Header einige Dinge am System beeinflusst werden können, sollten Sie "Vertraut" nur auf "Ja" setzen, wenn Sie genau wissen, von welchen Absendern die abgerufenen Nachrichten stammen. X-OTRS-Header werden vom Modul für die Nachrichtenfilterung in OTRS benutzt, die X-OTRS-Header werden in dieser Tabelle näher beschrieben. Eventuell eingerichtete Filterregeln werden auf jeden Fall abgearbeitet, egal welcher Wert für "Vertraut" ausgewählt wurde.

Weiterhin können Sie die Verteilung der abgerufenen Mails durch die Angabe steuern, ob die neuen Nachrichten nach dem To-Feld oder nach der Queue im System einsortiert werden sollen. Wählen Sie "Verteilung nach ausgewählter Queue" aus, landen die abgerufenen Mails auf jeden Fall in der Queue, die zusätzlich in der dafür vorgesehenen Listbox angegeben werden kann. Dabei spielt keine Rolle, an welche Adresse die Mail geschickt wurde. Wählen Sie "Verteilung nach To: Feld" aus, wird überprüft, welcher Queue die Adresse zugeordnet ist, an die die abgerufene Mail gesendet wurde. Die Zuordnung einer Mailadresse zu einer Queue kann über die Mailadressen Verwaltung vorgenommen werden. Existiert eine Zuordnung der Adresse im To: Feld zu einer Queue innerhalb des Systems, wird die abgerufene Nachricht in die entsprechende Queue einsortiert. Kann keine Zuordnung gefunden werden, landet das Ticket in der Standard-Queue des Systems (Raw), die mit Hilfe des Konfigurationsparameters PostmasterDefaultQueue eingestellt werden kann.

Die Daten zu allen Mail-Konten werden in der Datenbank von OTRS gespeichert. Das Skript `otrs.PostMasterMailbox.pl`, welches sich im Verzeichnis `bin` innerhalb des OTRS-Homeverzeichnisses befindet, fragt die Einstellungen ab und holt die Mails von den einzelnen Mail-Konten. Sie können dieses Skript manuell ausführen, um zu prüfen, ob alle E-Mail-Einstellungen korrekt funktionieren.

Auf einer Standardinstallation holt der Cron-Job `postmaster_mailbox` die E-Mails alle 10 Minuten ab. Das Kapitel "Einrichten der von OTRS benötigten cron-Jobs" beschreibt das Zusammenspiel zwischen OTRS und cron ausführlicher.

Anmerkung

Beim Abholen löscht OTRS die E-Mails vom POP oder IMAP server. Es besteht keine Möglichkeit, die Mails dort zu erhalten. Wenn Sie dies wünschen, erstellen Sie am Besten Weiterleitungsregeln auf Ihrem Mail-Server. Lesen Sie hierzu bitte in der Dokumentation ihres Mail-Servers nach.

Via Kommandozeilen-Programm und z. B. procmail (otrs.PostMaster.pl)

OTRS ist in der Lage, E-Mails über ein Kommandozeilen-Programm (`bin/otrs.PostMaster.pl`) zu empfangen. Es nimmt die Mails über STDIN entgegen und fügt sie direkt in das OTRS-System ein. Das bedeutet, dass E-Mails im OTRS angezeigt werden, wenn der MDA (mail delivery agent, z. B. procmail) die E-Mails an `bin/otrs.PostMaster.pl` weiterleitet.

Um `bin/otrs.PostMaster.pl` ohne MDA zu testen, führen Sie die Kommandos in Skript aus.

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

Skript: PostMaster-Test ohne MDA.

Wird die E-Mail in der Queue-Ansicht angezeigt, sind Ihre Einstellungen in Ordnung.

Procmail ist in der Linux-Umgebung ein sehr bekannter E-Mail-Filter, der höchstwahrscheinlich auf Ihrem System installiert sein wird. Falls nicht, erhalten Sie auf der *procmail Homepage* [<http://www.procmail.org/>] weitere Informationen.

Um procmail einzurichten (benötigt einen für procmail konfigurierten MDA (z. B. sendmail, postfix, exim oder qmail)), kann die Datei `.procmailrc.dist` aus dem OTRS-Homeverzeichnis verwendet werden. Kopieren Sie `.procmailrc.dist` nach `.procmailrc` und nehmen Sie die in Skript dargestellten Änderungen vor.

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

Skript: Konfiguration von procmail für OTRS.

Alle an den lokalen OTRS-Benutzer gesendeten E-Mails werden an `bin/otrs.PostMaster.pl` weitergeleitet und dadurch im Ticket-System gespeichert.

E-Mails via POP3 oder IMAP und fetchmail für otrs.PostMaster.pl empfangen

Um E-Mails von Ihrem Mailserver via POP3 oder IMAP für den OTRS-Rechner/lokalen OTRS-Benutzer und procmail abzuholen, benutzen Sie fetchmail [<http://fetchmail.berlios.de/>].

Anmerkung

Voraussetzung ist eine funktionierende SMTP-Konfiguration auf dem OTRS-Rechner.

Eine Beispielkonfiguration finden Sie in der Datei `.fetchmailrc.dist` im Homeverzeichnis von OTRS. Kopieren Sie diese Datei nach `.fetchmailrc` und erweitern Sie die Datei um die Daten Ihrer E-Mail-Accounts (siehe Beispiel 7-1).

Beispiel 7.1. `.fetchmailrc`

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

Vergessen Sie nicht, die Zugriffsrechte von `.fetchmailrc` auf 710 zu setzen.

Wird das Kommando "**fetchmail -a**" mit dieser Konfiguration ausgeführt, werden alle E-Mails auf das lokale OTRS-Konto weitergeleitet. Richten Sie ggf. einen Cronjob ein, wenn die E-Mails regelmäßig geholt werden sollen.

Filterung/Verteilung über PostMaster-Module (für komplexere Verteilungsszenarien)

Falls die `bin/otrs.PostMaster.pl` oder `bin/otrs.PostMasterMailbox.pl` Methoden verwendet werden, können X-OTRS-Header mit Hilfe der PostMaster-Filtermodule in die eingehenden Mails eingefügt bzw. bereits vorhandene X-OTRS-Header verändert werden. Mit Hilfe von X-OTRS-Headern kann das Ticket System bestimmte Aktionen für Mails ausführen, z. B. diese in eine bestimmte Queue einsortieren, sie einem bestimmten Kunden zuordnen, die Priorität ändern usw. Eine nähere Beschreibung der X-OTRS-Header finden Sie im Kapitel zum Einrichten von Mail-Accounts über den Administrations-Bereich von OTRS.

Es gibt verschiedene Standard-Filtermodule:

Anmerkung

Der Jobname (z. B. `$Self->{"PostMaster::PreFilterModule"}->{"Jobname"}`) muss eindeutig sein!

`Kernel::System::PostMaster::Filter::Match` ist ein Standard-Modul, um einige E-Mail-Header (z. B. From, To, Subject) zu prüfen und dann den neuen E-Mail-Header zu setzen (z. B. X-OTRS-Ignore: yes oder X-OTRS-Queue: spam).

Beispiel 7.2. Beispiel-Jobs für das Filtermodul `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 ist ein Standard-Modul, um die E-Mails an ein externes Kommando zu leiten. Dessen Ergebniss wird an STOUT ausgegeben und, falls das Ergebnis aus STDOUT true ist, wird ein neuer E-Mail-Header gesetzt (z. B. X-OTRS-Ignore: yes oder X-OTRS-Queue: spam).

Beispiel 7.3. Beispiel-Job für das Filtermodul 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',
    },
};
```

Natürlich ist es auch möglich, eigene PostMaster-Filtermodule zu entwickeln.

Kapitel 8. Zeitabhängige Funktionen in OTRS

Arbeitszeiten, Feiertage und Zeitzonen für das System festlegen

Einige Funktionen in OTRS, wie das automatische Entsperren von Tickets, setzen eine korrekte Konfiguration von Arbeitszeiten, Zeitzonen und Feiertagen voraus. Sie können diese in der SysConfig definieren, in Framework > Core::Time. Ebenso können Sie verschiedene Typen von Arbeitszeiten, Feiertagen und Zeitzonen als separate "Kalender" in Framework > Core::Time::Calendar1 bis Framework > Core::Time::Calendar9 definieren. Kalender können auf Queue- oder SLA-Ebene definiert werden. Das heißt, dass Sie z. B. einen Kalender mit 5x8 Arbeitsstunden pro Woche für Ihren "Standard"-SLA definieren können, zusätzlich jedoch auch einen mit 7x24 für Ihren "Gold"-SLA. Ebenso kann der Kalender für Ihre Queue "Support-USA" ein anderes Arbeitszeitfenster haben als ihre Queue "Support-Japan". OTRS kann bis zu 99 verschiedene Kalender verwalten.

Arbeitszeiten

Setzen Sie die Arbeitszeiten für Ihr System in der SysConfig fest, in Framework> Core::Time::TimeWorkingHours oder in der Konfiguration Ihrer Kalender. OTRS unterstützt eine Granularität von einer Stunde. Um z. B. 8:00 bis 18:00 als Arbeitszeit festzulegen, müssen die Auswahlkästchen 8, 9, 10 .. 17 aktiviert werden.

Nur während dieser Stunden können Tickets eskalieren, Benachrichtigungen zu Erinnerungstickets versendet oder Tickets automatisch freigegeben werden. Weiterhin werden auch nur diese Stunden in die Berechnung der Eskalationszeit und der Zeit für die automatische Freigabe mit einbezogen.

Feiertage mit festem Datum

Feiertage deren Datum jedes Jahr gleich ist, können dem System in TimeVacationDays oder im entsprechenden Abschnitt in den Kalendern 1-9 bekannt gemacht werden.

Während der hier festgelegten Tage werden keine zeitabhängigen Aktionen oder Berechnungen auf Tickets im System ausgeführt.

Anmerkung

In OTRS sind die *deutschen* Feiertage voreingestellt.

TimeVacationDaysOneTime

Freie oder Feiertage, für die sich jährlich das Datum ändert, können in TimeVacationDaysOneTime angegeben werden.

Während der hier festgelegten Tage werden keine zeitabhängigen Aktionen oder Berechnungen auf Tickets im System ausgeführt.

Anmerkung

OTRS hat für diese Feiertage keine Voreinstellung. Sie müssen diese, wie z. B. Ostern, dem System also selbst bekannt machen.

Automatische Ticket-"Freigabe" (unlock)

Gesperrte Tickets können automatisch vom System freigegeben werden. Diese Funktion kann z. B. dann nützlich sein, wenn sich ein Agent im Urlaub befindet und noch Tickets gesperrt hat, die bearbeitet werden sollen / müssen. Dieses Feature entsperrt Tickets nach einer bestimmten Zeit, um sicherzustellen, dass gesperrte Tickets nicht vergessen werden und andere Agenten darauf zugreifen können.

Die Zeit, nach der gesperrte Tickets automatisch freigegeben werden, kann in den Einstellungen jeder Queue festgelegt werden. Mit Hilfe des Moduls `bin/otrs.UnlockTickets.pl`, das als cron-Job regelmäßig ausgeführt werden sollte, wird die automatische Freigabe von Tickets umgesetzt.

Benachrichtigungen für entsperrte Tickets werden nur zu den Benutzern versendet, die die jeweilige Queue in "Meine Queues" eingetragen und die Benachrichtigung für entsperrte Tickets in ihren persönlichen Einstellungen aktiviert haben.

Tickets werden entsperrt wenn folgende Bedingungen zutreffen:

- In der Queue des Tickets ist ein *Freigabe-Zeitintervall* definiert.
- Das Ticket ist *gesperrt*.
- Der Status des Tickets ist *offen*.

Die Freigabezeit wird zurückgesetzt, wenn ein Agent einen neuen externen Artikel zum Ticket hinzufügt. Das kann einer der folgenden Artikeltypen sein: *email-external*, *phone*, *fax*, *sms*, or *note-external*.

Die Freigabezeit wird auch zurückgesetzt, wenn der letzte Artikel von einem Agent stammt und dann der Kunde einen hinzufügt, per E-Mail oder Weboberfläche.

Zuletzt wird die Freigabezeit auch dann zurückgesetzt, wenn das Ticket einem anderen Agenten zugewiesen wird.

Kapitel 9. Ticket-Verantwortlicher und Beobachten von Tickets

Seit OTRS 2.1 kann zusätzlich zum Besitzer ein Verantwortlicher für das Ticket festgelegt werden. Zudem können alle Aktivitäten rund um das Ticket beobachtet werden. Diese beiden Funktionalitäten sind mit den TicketResponsible- und dem TicketWatcher-Funktionen implementiert, und erleichtern das Zuweisen von Aufgaben und die Arbeit in hierarchischen Teamstrukturen.

Ticket-Verantwortlicher

Die Funktion "Verantwortlicher" eines Tickets macht es möglich, das Ticket einem anderen als dem aktuellen Besitzer zur Bearbeitung zu übergeben. Ein Benutzer, der ein Ticket gesperrt hat, kann es einem anderen Agenten, der nicht der Ticketbesitzer ist, zur Beantwortung übergeben. Nach der Bearbeitung kann der erste Benutzer die Ticketverantwortung vom zweiten wieder entfernen.

Mit dem Konfigurationsparameter `Ticket::Responsible` kann die Funktion TicketResponsible aktiviert werden. Dadurch werden drei neue Icons in ihrer Toolbar erscheinen.

Die Ticketverantwortung kann übergeben werden, indem das Ticket aufgerufen und die Funktion "Verantwortlicher" in der Aktionsleiste genutzt wird.

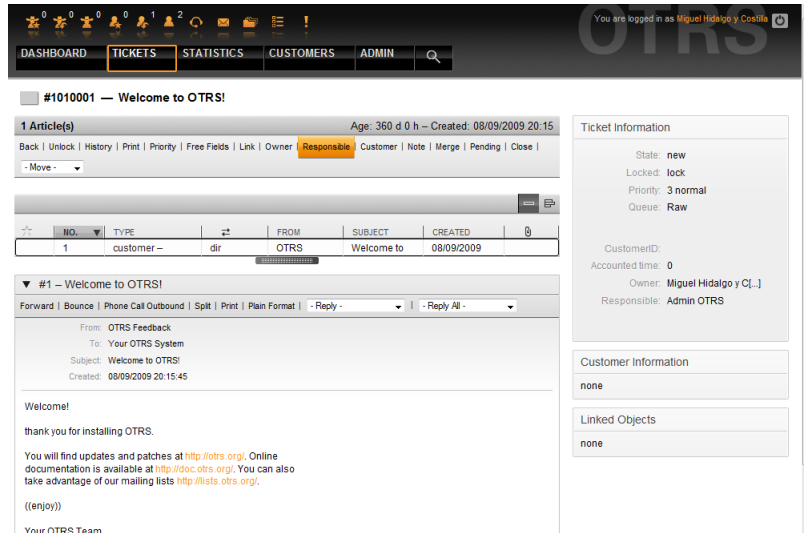


Abb.: Verantwortlichen eines Tickets ändern.

Nach dem Klick auf "Verantwortlicher" öffnet sich ein Popup-Fenster, in dem die Änderung durchgeführt werden kann. Damit kann auch dem neuen verantwortlichen Benutzer eine Nachricht geschickt werden.

Ticket-Verantwortlicher und Beobachten von Tickets

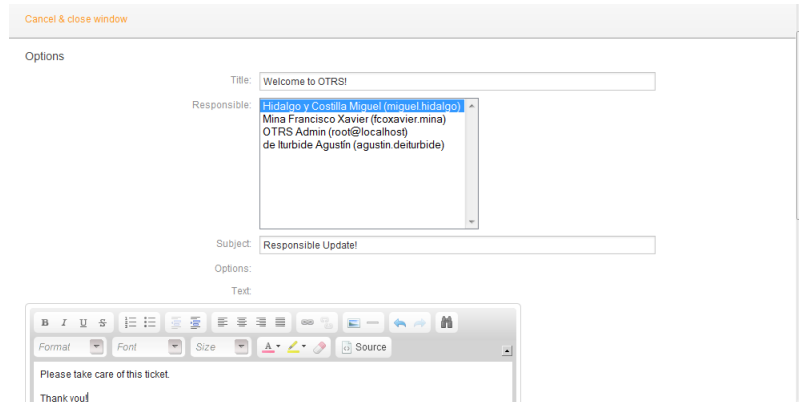


Abb.: Popup-Dialog zum Ändern des Ticket-Verantwortlichen.

Die Liste aller Tickets, für die ein Benutzer verantwortlich ist, kann durch die Ansicht "Verantwortliche Tickets" eingesehen werden. Diese kann über 3 Icons in der Toolbar aufgerufen werden, sobald das TicketResponsible-Feature aktiviert wird.

Tickets beobachten

Sollen innerhalb des Ticket-Systems lediglich bestimmte Tickets z. B. von einem Abteilungsleiter beobachtet und nicht selbst bearbeitet werden, so kann dies seit OTRS 2.1 mit Hilfe des TicketWatcher-Features realisiert werden.

Das TicketWatcher-Feature kann mit Hilfe des Konfigurations-Parameters Ticket::Watcher aktiviert werden. Weiterhin können über Ticket::WatcherGroup eine oder mehrere Benutzergruppen festgelegt werden, die die Erlaubnis haben sollen Tickets zu beobachten.

Um ein Ticket zu beobachten, muss der Inhalt des Tickets aufgerufen und im Menü für die verschiedenen Aktionen für das Ticket der "Beobachten"-Schalter betätigt werden.

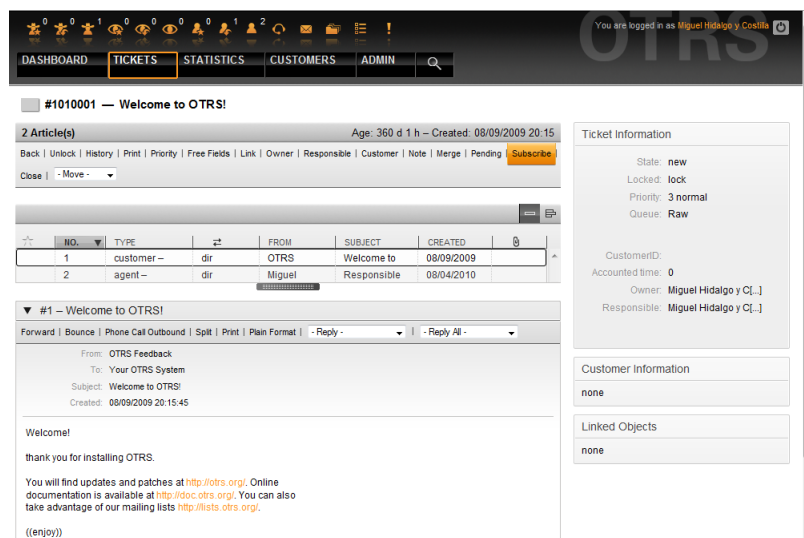


Abb.: Ticket beobachten.

Die Beobachtung eines Tickets wird aufgehoben, indem in der Inhaltsanzeige eines Tickets im Menü für die möglichen Aktionen der "Nicht beobachten"-Schalter gedrückt wird.

Ticket-Verantwortlicher und Beobachten von Tickets

0 0 1 0 0 1 0 1 2

You are logged in as Miguel Hidalgo y Costilla

DASHBOARD TICKETS STATISTICS CUSTOMERS ADMIN

#1010001 — Welcome to OTRS!

2 Article(s) Age: 360 d 2 h — Created: 08/09/2009 20:15

Back | Unlock | History | Print | Priority | Free Fields | Link | Owner | Responsible | Customer | Note | Merge | Pending |

Unsubscribe 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 Costilla
Responsible: Miguel Hidalgo y Costilla

Customer Information

none

Linked Objects

none

Abb.: Ticket nicht mehr beobachten.

Die Liste aller zur Beobachtung ausgewählten Tickets wird über die Ansicht "Beobachtete Tickets" erreicht. Diese kann über 3 Icons in der Toolbar aufgerufen werden, sobald das TicketWatcher-Feature aktiviert wird.

0 0 1 0 0 1 0 1 2

You are logged in as Miguel Hidalgo y Costilla

DASHBOARD TICKETS STATISTICS CUSTOMERS ADMIN

My Watched Tickets: All

All 1 | New Article 0 | Pending 0 | Reminder Reached 0

Bulk

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 Costilla	

1-1 of 1 - Page: 1 | S M L

Powered by OTRS 3.0.x CVS

Top of page

Abb.: Ansicht "Beobachtete Tickets".

Kapitel 10. Customizing the PDF output

Dieser Abschnitt behandelt die Einstellungen für die PDF-Druckausgabe von OTRS.

Wenn sie die Aktion "Drucken" in OTRS verwenden, wird eine formatierte PDF-Datei erzeugt. Sie können dies mit dem Konfigurationsparameter `PDF` deaktivieren. In diesem Fall wird stattdessen eine HTML-Ausgabe erzeugt.

Sie können das Erscheinungsbild der generierten Dateien anpassen, indem Sie ihr eigenes Logo in `PDF::LogoFile` hinterlegen. Mit `PDF::PageSize` lässt sich die Standard-Seitengröße der PDF-Datei einstellen. Schließlich kann mit `PDF::MaxPages` die maximale Anzahl an Seiten festgelegt werden, die eine PDF-Datei höchstens beinhalten soll. Das ist nützlich, um versehentliches Drucken überlanger Dokumente zu verhindern.

Für die Generierung von PDF-Dateien müssen die CPAN-Module `PDF::API2` und `Compress::Zlib` installiert werden. Bei vielen Distributionen sind diese als Paket verfügbar und können bequem über den jeweiligen Paketmanager installiert werden, falls dies jedoch nicht möglich ist muss der direkte Weg über CPAN gewählt werden. Das Kapitel Installation der für OTRS benötigten Perl-Module beschreibt, wie die Installation der Perl-Module durchzuführen ist.

Kapitel 11. Einbinden externer Backends

Kundenbenutzerdaten

OTRS ist in der Lage, mit verschiedenen Kundendaten (insbesondere Login, E-Mail, Telefon) umzugehen. Diese Informationen können im Agenten-Interface angezeigt und für das Kunden-Interface verwendet werden. Weiterhin werden die Daten für die Authentifizierung der Kunden am System benötigt.

Die benutzten/angezeigten Kundendaten sind frei konfigurierbar, es gibt jedoch folgende benötigte Optionen die unbedingt vorhanden sein müssen, damit OTRS ordnungsgemäß funktioniert:

- Benutzer-Login
- E-Mail-Adresse
- Benutzer-Kunden-ID

Verwenden Sie die Konfigurationsparameter aus dem Skript in Ihrer Datei `Kernel/Config.pm`, wenn Sie Kundeninformationen anzeigen möchten.

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

Skript: Konfigurationsparameter in `Kernel/Config.pm`.

Kundenbenutzer Backend

Es existieren zwei Kundenbenutzer Back-ends, DB und LDAP. Falls Sie bereits ein Kundenverzeichnis (z. B. SAP, ...) haben, ist es natürlich möglich, dafür ein eigenes Back-end zu schreiben.

Datenbank (Standard)

Beispiel 11-1 zeigt die Konfiguration eines Kundenbenutzer-Backends, welches die Kundendaten in der OTRS-Datenbank speichert.

Beispiel 11.1. Konfiguration eines Datenbank-Kunden-Backends

```
# CustomerUser (customer database backend and settings)
$Self->{'CustomerUser'} = {
```



```

    Name => 'Database Datasource',
    Module => 'Kernel::System::CustomerUser::DB',
    Params => {
        # if you want to use an external database, add the required
        settings
    #         DSN => 'DBI:odbc:yourdsn',
    #         DSN =>
    'DBI:mysql:database=customerdb;host=customerdbhost',
    #         User => '',
    #         Password => '',
    #         Table => 'customer_user',
    #         # 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.',
        },
    },
};

```

Falls Sie die Kundendaten anpassen möchten, ändern Sie in der Datenbank die Tabellenspalten oder fügen Sie weitere hinzu (im folgenden Beispiel wird ein Feld für die Raumnummer hinzugefügt).

```
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:~#
```

Skript: Ein Feld zur Tabelle customer_user hinzufügen.

Danach fügen Sie Ihre eigenen Spalten dem MAP Array in der Datei `Kernel/Config.pm` hinzu:

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

Skript: Hinzufügen eines Feldes in Kernel/Config.pm file.

Natürlich können Sie all diese Kundeninformationen dann auch über das Admin-Interface bzw. die Kundenverwaltung pflegen.

Kunden mit multiplen IDs (Firmen-Tickets)

Es ist möglich, einem Kunden mehr als nur eine Kundennummer zuzuweisen. Dies kann z. B. dann sinnvoll sein, wenn ein Kunde auf Tickets anderer Kunden zugreifen muss, z. B. der Abteilungsleiter auf die Tickets der Mitarbeiter seiner Abteilung. Hat ein Kunde Zugriff auf Tickets anderer Kunden, verwendet man in OTRS das sog. Firmen Ticket Feature. Im Kunden-Interface können diese Tickets über den "Firmen-Ticket" Link eingesehen werden.

Um Firmen-Tickets zu verwenden, muss die `customer_user` Tabelle in der OTRS Datenbank um eine Spalte erweitert werden, in die später die Kundennummern eingetragen werden, auf die ein Kunde zusätzlich zu den eigenen Tickets Zugriff haben soll.

```
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:~#
```

Skript: Hinzufügen eines Feldes customer_ids zur Tabelle customer_user.

Danach fügen Sie die neue Spalte dem MAP Array in der Datei `Kernel/Config.pm` hinzu.

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

Skript: Hinzufügen des Feldes UserCustomerIDs zu Kernel/Config.pm.

Die Spalte für die Multi-Kundennummern kann ab nun über das Admin-Interface bzw. über die Kundenverwaltung gepflegt werden.

Um nun den Zugriff für einen Kunden auf die Tickets anderer Kunden zu ermöglichen, tragen Sie in die neue Spalte die IDs der Kunden ein, auf deren Tickets der Zugriff ermöglicht werden soll. Die einzelnen IDs trennen Sie durch ein Semikolon.

Beispiel 11.2. Firmen-Tickets mit einem Datenbank-Backend

Angenommen es sind die Kunden A, B und C im System angelegt. A soll mit Hilfe von Firmen Tickets über das Kunden-Interface Zugriff auf die Tickets von B und C haben, B und C sollen jedoch jeweils nur ihre eigenen Tickets einsehen und bearbeiten können.

Um dieses Setup zu realisieren, ändern Sie wie oben beschrieben die `customer_user` Tabelle in der OTRS Datenbank und das Mapping in `Kernel/Config.pm`. Anschließend laden Sie über die Kundenverwaltung die Einstellungen des Kunden A und tragen bei "Kundennummern" die Werte "B;C;" ein.

LDAP

Falls Sie ein existierendes LDAP Verzeichnis mit Ihren Kundenbenutzern haben, können Sie dieses auch mit OTRS nutzen.

Beispiel 11.3. Konfiguration eines LDAP Kunden-Backends

```
# 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 ],
    ],
};

```

Falls Sie in Ihrem LDAP Verzeichnis weitere Informationen zu Ihren Kunden gespeichert haben und mit OTRS darauf zugreifen möchten, erweitern Sie das MAP Array in `Kernel/Config.pm` bzw. entfernen nicht gewünschte Einträge.

```

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

```

[ 'UserPhone',      'Phone',      'telephonenumber', 1, 0, 'var',
'', 0 ],

```

Skript: Hinzufügen neuer Felder zu Kernel/Config.pm.

Kunden mit multiplen IDs (Firmen-Tickets)

Um Firmentickets mit einem LDAP zu nutzen, muss ein Feld zum LDAP-Verzeichnis hinzugefügt werden, welches die IDs enthält, die für einen Kunden zugänglich sein sollen.

Wenn das neue Feld erstellt wurde, muss es zum MAP Array in `Kernel/Config.pm` hinzugefügt werden.

```

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

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

Skript: Hinzufügen eines neuen Feldes in Kernel/Config.pm.

Das Feld muss direkt im LDAP-Verzeichnis bearbeitet werden, weil OTRS von LDAP nur lesen kann, nicht jedoch darauf schreiben.

Um nun den Zugriff für einen Kunden auf die Tickets anderer Kunden zu ermöglichen, tragen Sie in das neue LDAP-Feld die IDs der Kunden ein, auf deren Tickets der Zugriff ermöglicht werden soll. Die einzelnen IDs trennen Sie durch ein Semikolon.

Beispiel 11.4. Verwendung von Firmentickets mit einem LDAP-Backend

Angenommen es sind die Kunden A, B und C im System angelegt. A soll mit Hilfe von Firmen Tickets über das Kunden-Interface Zugriff auf die Tickets von B und C haben, B und C sollen jedoch jeweils nur ihre eigenen Tickets einsehen und bearbeiten können.

Um dieses Setup zu realisieren, ändern Sie wie oben beschrieben das LDAP Verzeichnis und das Mapping in `Kernel/Config.pm`. Anschließend tragen Sie im LDAP Verzeichnis innerhalb der Einstellungen für den Kunden A für `CustomerIDs` die Werte "B;C;" ein.

Verwenden mehrerer Kunden Backends

Soll mehr als nur ein Back-end mit verschiedenen Kundendaten verwendet werden (z. B. gleichzeitig DB und LDAP), so ist dies ebenfalls mit OTRS möglich. In einem solchen Fall muss der `CustomerUser` Parameter für jedes Back-end um eine Nummer erweitert werden, z. B. "CustomerUser1", "CustomerUser2", usw.

Beispiel 11.5. Gleichzeitige Einbindung mehrerer verschiedener Kunden Backends

In der folgenden Konfiguration verwendet OTRS gleichzeitig ein Datenbank- und ein LDAP-Kunden-Backend.

```
# 1. Customer user backend: DB
# (customer database backend and settings)
$Self->{CustomerUser1} = {
    Name => 'Customer Database',
    Module => 'Kernel::System::CustomerUser::DB',
    Params => {
        # if you want to use an external database, add the
        # required settings
        DSN => 'DBI:odbc:yourdsn',
        DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',
        User => '',
        Password => '',
        Table => 'customer_user',
    },
    # customer unique id
    CustomerKey = 'login',
    # customer #
```

```

CustomerID = 'customer_id',
CustomerValid = 'valid_id',
CustomerUserListFields => ['first_name', 'last_name', 'email'],
CustomerUserSearchFields => ['login', 'last_name', 'customer_id'],
CustomerUserSearchPrefix => '',
CustomerUserSearchSuffix => '*',
CustomerUserSearchListLimit => 250,
CustomerUserPostMasterSearchFields => ['email'],
CustomerUserNameFields => ['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 ],
],
];
```

Es können bis zu 10 Kunden Back-ends gleichzeitig eingebunden werden. Über die Kundenverwaltung in OTRS ist der Zugriff auf die verschiedenen Back-ends möglich.

Backends für die Authentifizierung von Agenten und Kunden

OTRS bietet die Möglichkeit Agenten und Kunden über verschiedene Back-ends zu authentifizieren.

Authentifizierungs-Backends für Agenten

Datenbank (Standard)

Das Backend für die Authentifizierung von Agenten, welches OTRS standardmäßig verwendet, ist die OTRS-Datenbank. Die Agenten können innerhalb des Admin-Bereiches in der Benutzerverwaltung angelegt und bearbeitet werden.

Beispiel 11.6. Agentenauthentifizierung gegen ein Datenbank-Backend

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

LDAP

Falls ein LDAP Verzeichnis mit Ihren Agenten-Benutzerdaten verfügbar ist, können Sie das LDAP Modul für die Authentifizierung Ihrer Agenten nutzen. Dieses Modul greift nur lesend auf die Daten im LDAP Verzeichnis zu, d.h. die Daten können nicht mit OTRS bearbeitet werden, es können also keine Agenten mit Hilfe der Benutzerverwaltung von OTRS angelegt oder bearbeitet werden.

Beispiel 11.7. Agentenauthentifizierung gegen ein LDAP Backend

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

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

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

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

# in case you want to add a suffix to each login name, then
```

```
# you can use this option. e. g. user just want to use user but
# in your ldap directory exists user@domain.
#   $Self->{'AuthModule::LDAP::UserSuffix'} = '@domain.com';

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

Mit den folgenden Konfigurationsparametern können die Benutzerdaten der Agenten aus dem LDAP in die lokale OTRS Datenbank synchronisiert werden. Dies reduziert die Zugriffe auf ihr LDAP Verzeichnis, entlastet den Server mit den LDAP Daten und beschleunigt die Anmeldung an OTRS. Die Synchronisierung der Daten findet bei der ersten Anmeldung des Agenten statt, trotz der synchronisierten Daten bleibt ihr LDAP Verzeichnis die letzte Instanz bei der Anmeldung. D.h. wird ein User im LDAP Verzeichnis gelöscht oder deaktiviert, klappt die Anmeldung an OTRS nicht. Ebenfalls müssen die Daten für einen Agenten weiterhin direkt im LDAP Verzeichnis gepflegt werden.

```
# 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',
];
```

Skript: Synchronisation der Benutzerdaten aus dem LDAP-Verzeichnis in die OTRS-Datenbank.

HTTPBasicAuth für Agenten

Falls Sie eine "single sign on"-Lösung für Ihre Agenten implementieren möchten, benutzen Sie http basic authentication (für alle Ihre Systeme) und aktivieren Sie das HTTPBasicAuth Modul.

Beispiel 11.8. Agentenauthentifizierung über HTTPBasic

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

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

Radius

Mit den folgenden Einstellungen kann die Authentifizierung von Agenten gegen einen Radius-Server realisiert werden.

Beispiel 11.9. Agentenauthentifizierung gegen ein Radius-Backend

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

Authentifizierungs-Backends für Kunden

Datenbank (Standard)

Das Backend für die Authentifizierung von Kunden, welches OTRS standardmäßig verwendet, ist die OTRS-Datenbank. Die Kundendaten können über das Interface zur Verwaltung von Kunden angelegt und bearbeitet werden.

Beispiel 11.10. Kundenauthentifizierung gegen ein Datenbank-Backend

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

```

$$Self->{'Customer::AuthModule::DB::DSN'} =
    "DBI:mysql:database=customerdb;host=customerdbhost";
$$Self->{'Customer::AuthModule::DB::User'} = "some_user";
$$Self->{'Customer::AuthModule::DB::Password'} = "some_password";

```

LDAP

Falls ein LDAP Verzeichnis mit Ihren Kundenbenutzern verfügbar ist, können Sie das LDAP Modul für die Authentifizierung Ihrer Kunden nutzen. Dieses Modul greift nur lesend auf die Daten im LDAP Verzeichnis zu, d.h. die Daten können nicht mit OTRS bearbeitet werden, es können also keine Kunden mit Hilfe der Kundenverwaltung von OTRS angelegt oder bearbeitet werden.

Beispiel 11.11. Kundenauthentifizierung gegen ein LDAP Backend

```

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

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

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

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

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

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

```

```
version => 3,
};
```

HTTPBasicAuth für Kunden

Falls Sie eine "single sign on"-Lösung für Ihre Kunden implementieren möchten, benutzen Sie HTTPBasic Authentication (für alle Ihre Systeme) und aktivieren Sie das HTTPBasicAuth Modul (kein OTRS-Login mehr benötigt).

Beispiel 11.12. Kundenauthentifizierung über 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';
```

Radius

Mit den folgenden Einstellungen kann die Authentifizierung von Kunden gegen einen Radius-Server realisiert werden.

Beispiel 11.13. Kundenauthentifizierung gegen ein Radius-Backend

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

Kunden-Selbstregistrierung anpassen

Es ist möglich, die Kunden-Selbstregistrierung für neue Kunden über "customer.pl" anzupassen. Somit können Sie mehr optionale oder benötigte Felder (z. B. Adresse, Ort, Telefonnummer) hinzufügen.

In folgenden Beispiel wird ein benötigtes Feld für die Raumnummer hinzugefügt.

Anpassen der Weboberfläche

Damit im Webinterface das zusätzliche Feld für die Raumnummer angezeigt wird, muss die zuständige dtl-Datei angepasst werden. Editieren Sie `Kernel/Output/HTML/Standard/CustomerLogin.dtl` und fügen Sie das gewünschte Feld hinzu.

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

Skript: Anzeigen eines neuen Feldes in der Weboberfläche.

Kunden-Mapping

Zusätzlich muss das Kunden-Mapping um den Eintrag für die Raumnummer erweitert werden. Dazu werden zuerst die Einstellungen für "CustomerUser" aus der Datei `Kernel/Config/Defaults.pm` in die Datei `Kernel/Config.pm` übertragen. Anschließend wird das Kunden-Mapping um das Raumnummer-Feld erweitert.

```
# CustomerUser
# (customer database backend and settings)
$Self->{CustomerUser} = {
  Name => 'Database Backend',
  Module => 'Kernel::System::CustomerUser::DB',
  Params => {
    # if you want to use an external database, add the
    # required settings
    # DSN => 'DBI:odbc:yourdsn',
    # DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',
    # User => '',
    # Password => '',
    Table => 'customer_user',
  },
  # customer unique id
  CustomerKey => 'login',
  # customer #
  CustomerID => 'customer_id',
  CustomerValid => 'valid_id',
  CustomerUserListFields => ['first_name', 'last_name', 'email'],
  # CustomerUserListFields => ['login', 'first_name', 'last_name',
  'customer_id', 'email'],
  CustomerUserSearchFields => ['login', 'last_name', 'customer_id'],
  CustomerUserSearchPrefix => '',
  CustomerUserSearchSuffix => '*',
  CustomerUserSearchListLimit => 250,
  CustomerUserPostMasterSearchFields => ['email'],
  CustomerUserNameFields => ['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.',
        },
    },
};
```

Skript: Anpassung des MAP-Arrays.

Anpassen der Kunden-Tabelle in der OTRS-Datenbank

Abschließend muss eine neue Spalte zur "customer_user" Tabelle in der OTRS Datenbank hinzugefügt werden, in der die Raumnummer gespeichert werden kann.

```
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:~#
```

Skript: Hinzufügen einer neuen Spalte zur Tabelle customer_user.

Alle benötigten Anpassungen sind durchgeführt und das Feld für die Raumnummer sollte nun im Kunden-Interface (customer.pl) angezeigt und verwendet werden können. Neue Kunden müssen nun ihre Raumnummer angeben, wenn sie einen neuen Zugang einrichten wollen. Wird mod_perl eingesetzt, sollte der Webserver neu gestartet werden um die Änderungen zu übernehmen.

Kapitel 12. Status

Vordefinierte Ticket-Status

OTRS erlaubt es Ihnen, die Ticket-Status zu verändern oder neue Status hinzuzufügen. Hierbei gibt es zwei wichtige Optionen: den Namen des Status und Typ des Status.

Die standardmäßig voreingestellten Status lauten: 'neu', 'offen', 'erfolgreich geschlossen', 'erfolglos geschlossen', 'merged', 'entfernt', 'warten auf erfolgreich schließen', 'warten auf erfolglos schließen' und 'warten zur Erinnerung'.

Neu

Tickets sind normalerweise nur in diesem Status, wenn sie von eingehenden E-Mails neu erzeugt werden.

Öffnen

Der Standardstatus für Tickets, die Queues und Agents zugeordnet sind.

Warten zur Erinnerung

Nachdem die Wartezeit abgelaufen ist, erhält der Ticketbesitzer eine E-Mail zu dem Ticket. Wenn es nicht gesperrt ist, wird die Erinnerung an alle Agenten der Queue verschickt. Erinnerungs-Benachrichtigungen werden nur innerhalb der Arbeitszeiten versendet, und werden alle 24 Stunden wiederholt, bis der Status des Tickets vom Agenten verändert wird. Die Zeit, die dieses Ticket in diesem Status verbringt, wird dennoch bei der Berechnung der Eskalationszeiten einbezogen.

Warten auf erfolglos schließen

Wenn die Wartezeit abgelaufen ist, wird das Ticket auf "erfolglos geschlossen" gesetzt. Die Zeit, die dieses Ticket in diesem Status verbringt, wird dennoch bei der Berechnung der Eskalationszeiten einbezogen.

Warten auf erfolgreich schließen

Wenn die Wartezeit abgelaufen ist, wird das Ticket auf "erfolgreich geschlossen" gesetzt. Die Zeit, die dieses Ticket in diesem Status verbringt, wird dennoch bei der Berechnung der Eskalationszeiten einbezogen.

Zusammengefügt

Status von Tickets, die mit anderen Tickets zusammengefügt wurden.

Erfolgreich geschlossen

Endstatus für Tickets, die erfolgreich gelöst werden konnten. Je nach Konfiguration kann es unmöglich sein, geschlossene Tickets wieder zu eröffnen.

Erfolglos geschlossen

Endstatus für Tickets, die NICHT erfolgreich gelöst werden konnten. Je nach Konfiguration kann es unmöglich sein, geschlossene Tickets wieder zu eröffnen.

Anpassen von Ticket-Status

Jeder Status besteht aus einem Namen ("state-name") und einem Typen ("state-type"). Der Name ist frei wählbar und kann über das Admin-Interface von OTRS angepasst werden, die Statustypen müssen direkt in der Datenbank geändert werden. Letzteres wird jedoch nicht empfohlen und kann zu unerwarteten Ergebnissen führen. Z. B. hängen die Berechnung der Eskalationszeiten und auch die Funktion zum automatischen Entsperren von Tickets von bestimmten Statustypen ab.

Im Admin-Interface können Sie innerhalb der Einstellungen für "Status" neue Status für die vorhandenen Statustypen hinzufügen oder ändern. Beachten Sie, dass Sie bei Änderungen am Status "neu - new" auch die entsprechenden Änderungen in der Konfigurationsdatei `Kernel/Config.pm` bzw. mit Hilfe des grafischen Konfigurations-Front-End vornehmen müssen. Folgende Änderungen müssen für einen geänderten "neu"-Status vorgenommen werden:

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

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

Skript: Anpassung der Einstellungen von Kernel/Config.pm.

Möchten Sie einen neuen Statustyp hinzufügen, müssen Sie zuerst die `ticket_status-type`-Tabelle in der OTRS Datenbank mit Hilfe eines entsprechenden Datenbankclient anpassen.

```
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:~#
```

Skript: Anpassung der Statustypen.

Nun können Sie den neuen Statustyp verwenden. Nachdem ein Status mit diesem Statustyp verknüpft wurde, muss auch die OTRS-Konfiguration angepasst werden, damit der neue Status korrekt verwendet werden kann. Passen Sie die Folgenden Optionen in der SysConfig an:

Ticket -> Frontend::Agent::Ticket::ViewPhoneNew > AgentTicketPhone####StateDefault - um den Standardstatus für Telefontickets einzustellen.

Ticket -> Frontend::Agent::Ticket::ViewPhoneNew > AgentTicketPhone####StateType - um die verfügbaren Status für Telefontickets einzustellen.

Ticket -> Frontend::Agent::Ticket::ViewEmailNew > AgentTicketEmail####StateDefault - um den Standardstatus für E-Mail-Tickets einzustellen.

Ticket -> Frontend::Agent::Ticket::ViewEmailNew > AgentTicketEmail####StateType - um die verfügbaren Status für E-Mail-Tickets einzustellen.

Ticket -> Frontend::Agent::Ticket::ViewPhoneOutbound > AgentTicketPhoneOutbound####State - um den Standardstatus für neue Telefonartikel einzustellen.

Ticket -> Frontend::Agent::Ticket::ViewPhoneOutbound > AgentTicketPhoneOutbound####StateType - um die verfügbaren Status für neue Telefonartikel einzustellen

Ticket:Frontend::Agent::Ticket::ViewMove:Ticket::DefaultNextMoveStateType - um den Standardstatus für das Verschieben von Tickets einzustellen.

Ticket -> Frontend::Agent::Ticket::ViewBounce > StateDefault - um den Standardstatus für das Bouncen eines Tickets einzustellen

Ticket -> Frontend::Agent::Ticket::ViewBounce > StateType - um die verfügbaren Status für das Bouncen von Tickets einzustellen.

Ticket -> Frontend::Agent::Ticket::ViewBulk > StateDefault - um den Standardstatus in einer Sammelaktion einzustellen.

Ticket -> Frontend::Agent::Ticket::ViewBulk > StateType - um die verfügbaren Status in einer Sammelaktion einzustellen.

Ticket -> Frontend::Agent::Ticket::ViewClose > StateDefault - um den Standardstatus für das Schließen von Tickets einzustellen.

Ticket -> Frontend::Agent::Ticket::ViewClose > StateType - um die verfügbaren Status für das Schließen von Tickets einzustellen.

Ticket -> Frontend::Agent::Ticket::ViewCompose > StateDefault - um den Standardstatus für Antworten einzustellen.

Ticket -> Frontend::Agent::Ticket::ViewCompose > StateType - um die verfügbaren Status für das Antworten einzustellen.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateDefault - um den Standardstatus für das Weiterleiten von Tickets einzustellen.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateType - um die verfügbaren Status für das Weiterleiten von Tickets einzustellen.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateDefault - um den Standardstatus für das Bearbeiten der Freitextfelder festzulegen.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateType - um die verfügbaren Status für das Bearbeiten der Freitextfelder festzulegen.

Ticket -> Core::PostMaster > PostmasterDefaultState - um den Status von Tickets festzulegen, die aus E-Mails neu erzeugt werden.

Ticket -> Core::PostMaster > PostmasterFollowUpState - um den Status von Tickets festzulegen, nachdem ein Followup eingegangen ist.

Ticket -> Core::PostMaster > PostmasterFollowUpStateClosed - um den Status von bereits geschlossenen Tickets festzulegen, nachdem ein Followup eingegangen ist.

Ticket -> Core::Ticket > ViewableStateType - um die Ticketstatus festzulegen, die in verschiedenen Bereichen des Systems zur Anzeige kommen, z. B. in der Queue-Ansicht.

Ticket -> Core::Ticket > UnlockStateType - um die verfügbaren Status für das Entsperren von Tickets einzustellen.

Ticket -> Core::Ticket > PendingReminderStateType - um die Status für Erinnerungstickets festzulegen.

Ticket -> Core::Ticket > PendingAutoStateType - um die Status für Tickets mit "Warten auto" festzulegen.

Ticket -> Core::Ticket > StateAfterPending - um den Status von Tickets mit "Warten auto" einzustellen, deren Wartezeit abgelaufen ist.

Kapitel 13. Anpassen der Ticket-Prioritäten

OTRS enthält 5 vordefinierte Ticketprioritäten, die mit dem "Prioritäten"-Modul im Administrationsbereich verwaltet werden können. Wenn Sie eine angepasste Prioritätenliste erstellen, bedenken Sie bitte, dass diese in der Prioritätsauswahl-Liste alphabetisch sortiert dargestellt werden. In der Queue-Ansicht ordnet OTRS die Prioritäten nach internen Datenbank-IDs.

Anmerkung

In OTRS können Prioritäten deaktiviert, aber nicht gelöscht werden. Wenn Sie eine Priorität deaktivieren wollen, setzen Sie den Wert für "Gültig" auf "ungültig" oder "ungültig-temporär".

Wichtig

Wenn eine Priorität hinzugefügt oder geändert wurde, müssen Sie ggf. einige Werte in der SysConfig anpassen:

- Ticket:Core::Postmaster::PostmasterDefaultPriority - definiert die Standardpriorität für einkommende E-Mails.
- Ticket:Frontend::Agent:Ticket::ViewPhoneNew:Priority - definiert die Standardpriorität für neue Telefontickets im Agentenbereich.
- Ticket:Frontend::Agent:Ticket::ViewEmailNew:Priority - definiert die Standardpriorität für neue E-Mail-Tickets im Agentenbereich.
- Ticket:Frontend::Customer:Ticket::ViewNew:PriorityDefault - definiert die Standardpriorität für neue Tickets im Kundenbereich.

Kapitel 14. Erstellung eigener Themes

Für OTRS können verschiedene Themes angelegt werden, also verschiedene Layouts zur Gestaltung der Web-Oberfläche. Dazu müssen Sie die vorhandenen Templates ändern und Ihren Wünschen entsprechend anpassen.

More information on the syntax and structure of output templates can be found in the Developer Manual at <http://doc.otrs.org>, especially in the chapter on *templates* [<http://doc.otrs.org/developer/3.1/en/html/hacking.html#TemplatingMechanism>].

Führen Sie beispielsweise folgende Schritte durch um ein neues Theme namens "Company" zu erstellen:

1. Erstellen Sie das Verzeichnis `Kernel/Output/HTML/Company` und kopieren Sie die Dateien, die Sie für das Company-Theme anpassen möchten, aus `Kernel/Output/HTML/Standard` in das neu erstellte Verzeichnis.

Wichtig

Kopieren Sie nur diejenigen Dateien, die Sie verändern möchten. Fehlende Dateien wird OTRS automatisch vom Standard-Theme laden. Dadurch werden Systemaktualisierungen vereinfacht.

2. Passen Sie die Dateien im Verzeichnis `Kernel/Output/HTML/Company` Ihren Wünschen entsprechend an.
3. Um das neue Theme OTRS bekannt zu machen, fügen Sie es unter `Frontend::Themes` zur `SysConfig` hinzu.

Ab nun sollten sie das neue Theme nutzen und über Ihre persönlichen Einstellungen aktivieren können.

Warnung

Nehmen Sie keine Änderungen an den Theme-Dateien vor, die mit OTRS installiert wurden, da diese bei einem Update verloren gehen können. Erstellen Sie eigene Themes ausschließlich nach dem oben beschriebenen Verfahren.

Kapitel 15. Übersetzung des OTRS-Frontends

OTRS bietet Mehrsprachigkeit in der Weboberfläche an.

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" [<http://doc.otrs.org/developer/3.1/en/html/contributing.html#translate>] chapter from the developer manual on <http://doc.otrs.org>.

Kapitel 16. PGP

In OTRS können ausgehende E-Mails mit Hilfe von PGP signiert oder verschlüsselt werden. Ebenfalls ist es möglich verschlüsselte Nachrichten zu entschlüsseln. Die Ver- und Entschlüsselung mit PGP wird mit Hilfe des GPL-Werkzeugs GnuPG vorgenommen. Zur Einrichtung sind die folgenden Schritte notwendig:

1. Erste Aufgabe ist es, das entsprechende GnuPG-Software-Paket zu installieren, welches bei den meisten Linux-Distributionen mitgeliefert wird. Dies sollte mit Hilfe des jeweiligen Paketmanagers leicht durchgeführt werden können.
2. Im zweiten Schritt muss das soeben installierte GnuPG zur Benutzung für OTRS konfiguriert werden. Dies geschieht auf der Kommandozeilenebene durch einen Aufruf von GnuPG, der die notwendigen Verzeichnisse anlegt und den privaten Schlüssel erzeugt. Der Aufruf muss als Benutzer 'otrs' durchgeführt werden.

```
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
```

from the Real Name, Comment and Email Address in this form:
 "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

```
You selected this USER-ID:
"Ticket System (Private PGP Key for the ticket system with
dress support@examp
le.com) <support@example.com>"
```

```
Passphrase: secret
Repeat passphrase: secret
```

```

+++++++ .+++++++ . . . .+++++ .+++++ . . .+++++++
+++++++ .
+++++++ .+++++++ .+++++ .+++++ .+++++++>
+++++++>+++++
. . . . .>+++++<+++++ . . . . .+++++

```

the OS a chance to collect more entropy! (Need 280 more bytes)

.....++++^ ^ ^

```
gpg: checking the trustdb
gpg: 3 marginal(s) needed, 1 complete(s) needed, PGP trust model
gpg: depth: 0  valid:   1  signed:   0  trust: 0-, 0q, 0n, 0m,
    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:~$
```

Skript: Konfiguration von GnuPG.

Wie man sehen kann, genügt es bei den meisten Fragen die Vorgabe mit <EINGABE> zu bestätigen. Lediglich die Angabe zur Person des Schlüsselbesitzers ist zu vervollständigen bzw. zum Ende hin ist an der mit (passphrase) gekennzeichneten Stelle die Passphrase für den zu generierenden Schlüssel einzugeben.

3. Im nächsten Schritt muss OTRS auf die Verwendung von PGP vorbereitet werden. Suchen Sie in SysConfig nach "PGP" und wählen Sie danach die Untergruppe Crypt::PGP aus.

In der nun angezeigten Maske sollte zum einen " PGP aktiviert werden (die erste Option). Danach sollte überprüft werden, ob der voreingestellte Pfad zum Programm gpg der tatsächlichen Installation entspricht.

Die nächste Einstellung (PGP::Options) muss ggf. modifiziert werden. Es handelt sich um die Parameter, mit denen OTRS das Programm gpg aufruft. Hier ist insbesondere die Option für die Lage des GnuPG-Konfigurationsverzeichnis des OTRS-Benutzers otrs wichtig. Im Beispiel ist dies: /opt/otrs/.gnupg). Dieses Verzeichnis wurde im Schritt 1 automatisch von GnuPG angelegt.

Über die letzte Option können die Schlüssel-Werte-Paare für die ID(s) und Passphrase(n) der eigenen PGP-Schlüssel dem Ticket System bekannt gemacht werden. Noch einmal genauer: da andere Kommunikationspartner an das Ticket-System (oder besser: an den Mail-Eingang des Systems) E-Mails mit dessen öffentlichen Schlüssel verschlüsselt schicken, kann OTRS mit dem/n in dieser Option angegebenen privaten Schlüssel(n) solchermaßen verschlüsselte Mails entschlüsseln.

Woher bekommt man die ID des eigenen Schlüssels? Die ID steckt schon in der Ausgabe der Schlüsselgenerierung (siehe Schritt 1). Man kann die ID aber auch als Benutzer 'otrs' jederzeit über die Kommandozeile ermitteln:

```
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:~$
```

Skript: Die ID des eigenen privaten Schlüssels ermitteln.

Die ID des Schlüssels befindet sich in der Zeile, die mit sub beginnt und ist eine 8-stellige hexadezimale Kennung (im Beispiel lautet sie "52B97069". Die für die Option geforderte Passphrase ist dieselbe, die beim Schlüsselgenerieren in Schritt 1 verwendet wurde.

Nach Eingabe all dieser Angaben können sie mit dem Aktualisieren-Button gespeichert werden. OTRS ist jetzt für das Empfangen mit PGP verschlüsselter E-Mails konfiguriert.

4. Der letzte Schritt ist der Import des öffentlichen PGP-Schlüssels eines Kunden. Dadurch wird sicher gestellt, dass aus OTRS heraus verschlüsselte Mails an den jeweiligen Kunden gesendet werden können. Es bestehen zwei Möglichkeiten für den Import.

Zum einem können über das Modul zur Verwaltung der Kunden die öffentlichen PGP-Schlüssel beim Anlegen/Bearbeiten des jeweiligen Kunden im System hinterlegt werden.

Die zweite Möglichkeit bietet das System in den PGP-Einstellungen innerhalb des Admin-Bereiches. In dieser Konfigurationsmaske ist im rechten Teil eine Liste der bereits dem System zur Verfügung stehenden öffentlichen Schlüssel zu sehen. Im Regelfall sollte nach der obigen Einrichtung hier schon der öffentliche Schlüssel des Ticket-Systems selbst zu sehen sein. Im linken Teil besteht neben der Suche die Möglichkeit, neue Schlüssel als Schlüsseldatei ins System zu laden.

Sowohl für das Hinzufügen eines Schlüssels über die Kunden- als auch über die PGP-Verwaltung gilt, dass die Dateien mit den Schlüsseln PGP/GnuPG-konforme Schlüssel sein müssen. Normalerweise wird der Schlüssel als "ASCII armored key"-Datei vorliegen, welches problemlos von OTRS verarbeitet werden kann.

Kapitel 17. S/MIME

Die Einrichtung der Verschlüsselung mit S/MIME scheint auf den ersten Blick ein bisschen komplizierter als die PGP-Einrichtung zu sein, da für das OTRS-System erst einmal eine Certification Authority (CA) eingerichtet werden muss. Ansonsten ist das Vorgehen ähnlich wie bei PGP, OTRS konfigurieren, eigenes Zertifikat einrichten, ggf. fremde Public-Zertifikate importieren, usw.

Die S/MIME-Konfiguration geschieht zu einem großen Teil außerhalb der OTRS-Web-Oberfläche und sollte als otrs-Benutzer (bzw. als der Benutzer mit dessen Rechten OTRS läuft) in einer Shell durchgeführt werden. Da die MIME-Konfiguration unter Linux im wesentlichen auf SSL (openssl basiert, sollte zuerst sichergestellt werden, dass das openssl-Paket installiert ist. Mit dem openssl-Paket kommt ein Skript, `CA.pl`, mit welchem die wichtigsten Schritte zur Zertifikatserstellung bewältigt werden können. Damit dieser Vorgang einfacher wird, sollte zuerst herausgefunden werden, wo sich das Skript `CA.pl` im Dateisystem befindet. Danach sollte diese Stelle der Einfachheit halber temporär in den Suchpfad der Shell übernommen werden.

```
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>
```

Skript: Konfiguration von S/MIME.

Im Beispiel sieht man auch, dass ein temporäres Verzeichnis `~/tmp` angelegt wurde, in welchem die Zertifikatsgenerierung durchgeführt wird.

Im einzelnen sind zur Zertifikatserzeugung folgende Schritte durchzuführen, die einzelnen Schritte sind in der Kommandozeile auszuführen. Sollte ein beglaubigtes SSL-Zertifikat für die Verschlüsselung bereits vorhanden sein, so sollte natürlich dieses verwendet werden. Dann können die nun folgenden Schritte übersprungen werden. Der beschriebene Ablauf geht davon aus, dass der OTRS-Administrator sich das SSL-Zertifikat zu Test- und Lernzwecken selbst anlegen muss.

1. Anlegen einer eigenen Certification Authority für SSL. Diese wird benötigt, um die Anfrage für ein eigenes SSL-Zertifikat zu beglaubigen.

```
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>
```

Skript: Anlegen einer Certification Authority für SSL.

2. Erzeugen der Zertifikatsanfrage.

```
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>
```

Skript: Erstellen einer Zertifikatsanfrage.

3. Die Zertifikatsanfrage durch die CA signieren lassen. Die Zertifikatsanfrage kann entweder durch die selbst angelegte CA signiert (= beglaubigt) werden. Allerdings ist es natürlich seriöser, wenn das eigene SSL-Zertifikat von einer fremden, externen und ihrerseits beglaubigten CA beglaubigt wird.

```
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>
```

Skript: Signieren der Zertifikatsanfrage.

4. Mit der signierten Zertifikatsanfrage das eigene Zertifikat und alle dazugehörigen Dateien erzeugen.

```
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>
```

Skript: Neues Zertifikat erstellen.

Nach der Durchführung dieser Schritte ist es notwendig die Einrichtung von S/MIME in OTRS abzuschließen.

Die OTRS-seitige Einrichtung erfolgt aus dem Admin-Bereich, Block System über den Punkt SMIME. Falls die generelle S/MIME-Unterstützung in OTRS noch nicht aktiviert wurde, weist die Maske den OTRS-Administrator beim Aufruf darauf hin und bietet einen bequemen Link zur Einrichtung.

Über die SysConfig kann die generelle S/MIME-Unterstützung eingeschaltet und konfiguriert werden. Diese Einrichtung findet man unter der SysConfig-Gruppe Crypt::SMIME.

Hier können sie S/MIME-Unterstützung aktivieren, und die Pfade zum OpenSSL-Kommando und zu den Zertifikaten angeben. Die vorhin erzeugte Schlüsseldatei muss in diesem Verzeichnis gespeichert werden, sonst kann OpenSSL sie nicht verwenden.

Im nächsten Schritt geht es (zurück) zur S/MIME-Konfiguration im Administrationsbereich. Dort können nun sowohl der private Schlüssel(Key) bzw. die privaten Schlüssel des OTRS-Systems,

als auch die öffentlichen Schlüssel anderer Kommunikationspartner ins System importiert werden. Tragen Sie hier den öffentlichen Schlüssel ein, der zu Beginn dieses Abschnitts erzeugt und dann in OTRS hinzugefügt wurde.

Selbstverständlich können aber auch alle öffentlichen S/MIME-Schlüssel der Kommunikationspartner über das Modul zur Kundenverwaltung beim Anlegen/Bearbeiten des jeweiligen Kunden in das System importiert werden.

Kapitel 18. Access Control Lists (ACLs)

Introduction

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 through the process of defining ACL definitions, and a reference of all possible important ACL settings.

Examples

Beispiel 18.1. ACL, die das Verschieben von Tickets in eine Queue nur bei einer Ticket-Priorität von 5 erlaubt.

This example shows you the basic structure of an ACL. First, it needs to have a name. In this case, it is "ACL-Name-2". Note that the ACLs will be numerically sorted before execution, so you should use the names carefully.

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".

Lastly, a section "Possible" defines modifications to the screens. In this case, from the available queues, only the queue "Alert" can be selected in a ticket screen.

```
# 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'],
        },
    },
};
```

Beispiel 18.2. ACL, die das Schließen von Tickets in der raw-Queue verbietet und den Schließen-Schalter ausblendet.

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
    },
  },
};
```

Beispiel 18.3. 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'],
    },
  },
};
```

Beispiel 18.4. ACL only showing Hardware services for tickets that are created in queues that start with "HW".

This example also shows you how you can use regular expressions for matching tickets and for filtering the available options.

```
$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)'],
    },
  },
};
```

Reference

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

Beispiel 18.5. Reference showing all possible important ACL settings.

```
# ticket acl
$Self->{TicketAcl}->{'200-ACL-Reference'} = {
    # match properties
    Properties => {
        # current action match properties
        Frontend => {
            Action => ['AgentTicketPhone', 'AgentTicketEmail'],
        },
        # current queue match properties
        Queue => {
            Name      => ['Raw'],
            QueueID   => ['some id'],
            GroupID   => ['some id'],
            Email     => ['some email'],
            RealName  => ['OTRS System'],
            # ...
        },
    },
    # current user match properties
    User => {
        UserLogin => ['some login'],
        # ...
        Group_rw => [
            'hotline',
        ],
        # ...
    },
    # current customer user match properties
    CustomerUser => {
        UserLogin => ['some login'],
        # ...
    },
    # current service match properties
    Service => {
        ServiceID => ['some id'],
        Name      => ['some name'],
        ParentID  => ['some id'],
        # ...
    },
    # current type match properties
    Type => {
        ID      => ['some id'],
        Name    => ['some name'],
        # ...
    },
    # current priority match properties
    Priority = {
        ID      => ['some id'],
        Name    => ['some name'],
        # ...
    },
    # current SLA match properties
    SLA = {
        SLAID   => ['some id'],
```

```

        Name      => ['some name'],
        Calendar => ['some calendar'],
        # ...
    },
    # current state match properties
    State = {
        ID          => ['some id'],
        Name        => ['some name'],
        TypeName    => ['some state type name'],,
        TypeID      => ['some state type id'],
        # ...
    },
    # current ticket owner match properties
    Owner => {
        UserLogin => ['some login'],
        # ...
        Group_rw => [
            'some group',
        ],
        # ...
    },
    # current ticket responsible match properties
    Responsible => {
        UserLogin => ['some login'],
        # ...
        Group_rw => [
            'some group',
        ],
        # ...
    },
    # current dynamic field match properties
    DynamicField => {
        # keys must be in DynamicField_<field_name> format
        DynamicField_Field1      => ['some value'],
        DynamicField_OtherField  => ['some value'],
        DynamicField_TicketFreeText2 => ['some value'],
        # ...
    },
    # current ticket match properties
    Ticket => {
        Queue          => ['Raw'],
        State          => ['new', 'open'],
        Priority        => ['some priority'],
        Lock            => ['lock'],
        CustomerID      => ['some id'],
        CustomerUserID  => ['some id'],
        Owner           => ['some owner'],
        DynamicField_Field1 => ['some value'],      # Must be the
untranslated values                                     # specified
                                                         # definition
in the dynamic field
and not the IDs
        DynamicField_MyField => ['some value'],

```

```

        # ...
    },
},
# 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'],
        # ...
    },
    # 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"
    # ...
},
};

```


Kapitel 19. Statistikmodul

Das OTRS-Statistikmodul enthält Funktionen, um betriebliche Kennziffern zu überwachen und angepasste Reports der Verwendung von OTRS zu erstellen. In OTRS wird der Begriff "Stat/ Statistik" für einen Report benutzt, der verschiedene Indikatoren abbildet.

Eine sinnvolle Konfiguration des OTRS-Statistikmoduls ist verbunden mit einer Vielzahl von Anforderungen und Überlegungen. Darin eingeschlossen sind die auszuwertenden Module von OTRS, Benutzerrechte, zu berechnende Indikatoren und ihre Komplexitätsgrade, eine einfache Konfiguration des Statistikmoduls, Geschwindigkeit und Effizienz der Berechnungen, und die Unterstützung verschiedener Ausgabevarianten.

Um komplexe Statistiken abzubilden, besteht ferner die Möglichkeit statische Elemente zu integrieren (Dateien, die die Funktionalität des Statistik-Moduls um ganz spezifische Anforderungen ergänzen).

Bedienung des Moduls durch den Agent

Wenn man als Agent angemeldet ist, erscheint der Menüpunkt "Statistik" mit verschiedenen Unterpunkten, siehe Abb.

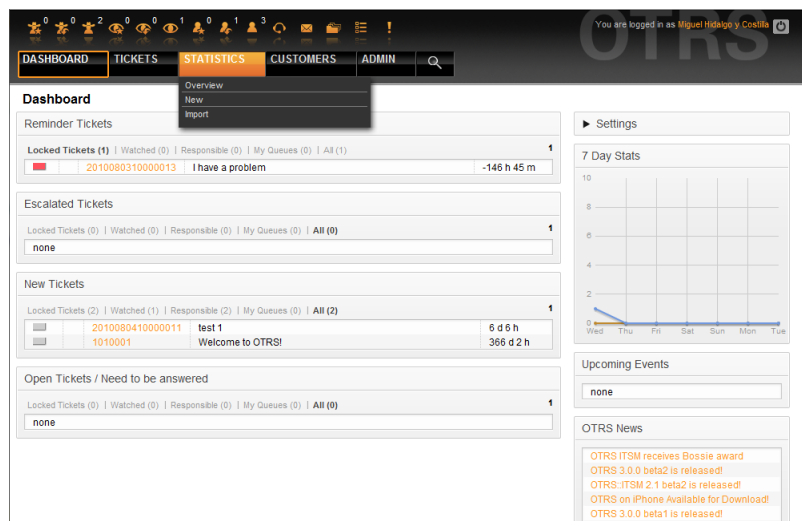


Abb.: Statistikmenü.

Folgende Optionen sind im Statistikmenü verfügbar:

- *Übersicht.* Zeigt eine Liste vorkonfigurierter Reports an.
- *Neu.* Benötigt "rw"-Rechte.
- *Importieren.* Benötigt "rw"-Rechte.

Übersicht

Die Auswahl von "Statistik" und des Unterpunktes "Übersicht" ruft den Übersichtsbildschirm auf. Hier wird dem Agenten eine Liste von vorkonfigurierten Reports präsentiert, die er verwenden kann.

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

Abb.: Übersicht der Standardstatistiken.

Folgende Informationen sind den in der Übersicht aufgelisteten Statistiken zu entnehmen:

- **Stat#.** Eindeutige Statistiknummer.
- **Titel.** Titel der Statistik.
- **Objekt.** Objekt, das für die Generierung der Statistik verwendet wird. Wenn es sich um eine statische Statistik handelt, wird kein Objekt angezeigt, da kein dynamisches Objekt zur Generierung verwendet wird.
- **Beschreibung.** Ein Auszug aus der Beschreibung der Statistik.

Wenn das Statistikmodul installiert ist, enthält es einige vorkonfigurierte Reports. Diese können in der Übersicht aufgerufen werden. Wenn die Liste zu lang für eine Seite wird, kann der Agent durch die verschiedenen Seiten blättern. Die Liste kann durch klick auf den Spaltenkopf wie gewünscht geordnet werden. Um einen bestimmten Report zu erzeugen, klicken Sie auf die Statistiknummer des Reports in der Übersicht. Dadurch gelangen Sie in die Ansichtsseite des Reports.

Ansehen und Generieren von Statistiken

Aufgabe der "Ansehen"-Oberfläche ist es, dem Agent einen Überblick über die Vorkonfiguration der Statistik zu ermöglichen.

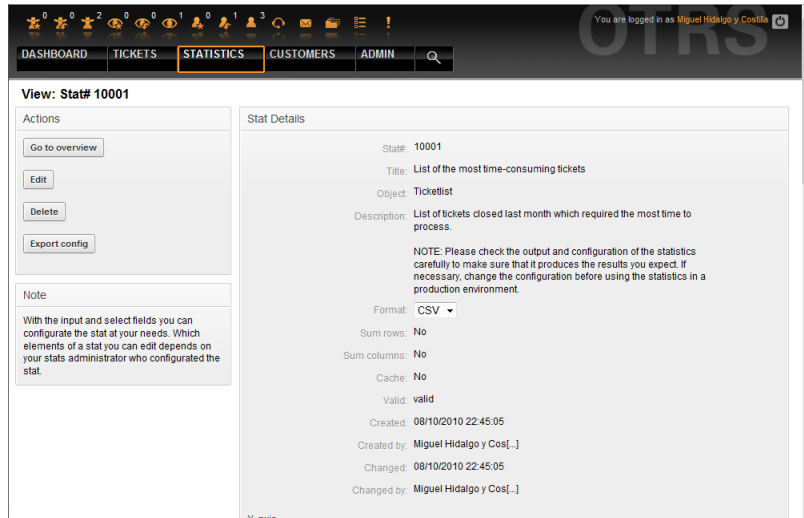


Abb.: Ansicht einer Statistik.

Konfigurationseinstellungen für einen Report können in den Optionen der Ansichtseite eingestellt werden. Der Ersteller des Reports und jeder Agent mit den benötigten Rechten kann diese Einstellungen vornehmen.

Konkret wird auf dieser Seite folgendes angezeigt:

- **Verfügbare Aktionen:**
 - *Zurück zur Übersicht.* Zurück zur Übersichtsliste der Reports.
 - *Bearbeiten.* Die aktuelle Statistik bearbeiten (benötigt "rw"-Rechte).
 - *Löschen.* Die aktuelle Statistik löschen (benötigt "rw"-Rechte).
 - *Konfiguration exportieren.* Exportieren einer Statistikkonfiguration mittels Dateidownload ("rw"-Rechte erforderlich).

Verwendung: Mittels der Export- und Importfunktionen können Statistiken bequem auf Testsystemen entworfen und getestet werden, um dann anschließend einfach in das Produktivsystem integriert zu werden.

- **Report-Details:**
 - *Statistik-Nr..* Die Nummer der Statistik.
 - *Titel.* Titel der Statistik.
 - *Objekt.* Objekt, das für die Generierung der Statistik verwendet wird.
 - *Beschreibung.* Ein Auszug aus der Beschreibung der Statistik.
 - *Format.* Ausgabeformat der Statistik. Hierbei stehen je nach Konfiguration folgende Ausgabeformate zur Verfügung:
 - CSV.
 - Drucken.

- Liniendiagramm.
- Balkendiagramm.
- Balkendiagramm (horizontal).
- Punktdiagramm
- Linienpunktdiagramm.
- Flächendiagramm.
- Tortendiagramm.
- *Grafikgröße*. Mögliche Größe der Grafik bzw. des Diagramms. Diese Auswahl bzw. Anzeige erscheint nur, wenn die Statistikvorkonfiguration ein Diagramm vorsieht bzw. ermöglicht. Alle generell verwendbaren Grafikgrößen werden durch den otrs-Admin in der SysConfig definiert. Beim Konfigurieren der Statistiken kann dann der Ersteller alle relevanten Formate vorselektieren.
- *Zeilensummierung*. Gibt an, ob die Statistik durch eine Spalte ergänzt wird, deren Zellen die Summe der jeweiligen Zeile darstellt.
- *Spaltensummierung*. Angabe, ob die Statistik durch eine Zeile ergänzt wird, deren Zellen die Summe der jeweiligen Spalte beinhaltet.
- *Cache*. Gibt an, ob die errechneten Statistiken im Dateisystem gecacht werden.
- *Gültig*. Ist eine vorkonfigurierte Statistik erst im Entstehen oder darf sie aus bestimmten Gründen temporär nicht mehr ausgeführt werden so wird das Element "Gültig" auf "ungültig" gesetzt. Ferner wird dadurch die Schaltfläche "Start" rechts unten am Ende des Block ausgeblendet. Ein Generieren der Statistik ist somit nicht möglich.
- *Erstellt*. Erstellungszeit der Statistik.
- *Erstellt von*. Agent, der die Statistik erstellt hat.
- *Geändert*. Letzte Änderungszeit der Statistik.
- *Geändert von*. Agent, der die Statistik zuletzt geändert hat.
- *X-Achse*. Diese Option erlaubt es dem Agenten, die X- und Y-Achsen zu vertauschen (Diese Funktion muss vom OTRS-Administrator freigeschaltet werden).
- Nach den allgemeinen Angaben sieht der Agent die Angaben zur eigentlichen Statistik. Hierbei kommen grundsätzlich zwei verschiedene Anzeigen zum Einsatz:
 - *Anzeige statischer Statistiken*. Statische Report-Generatoren können in die Statistik integriert werden (siehe unten).

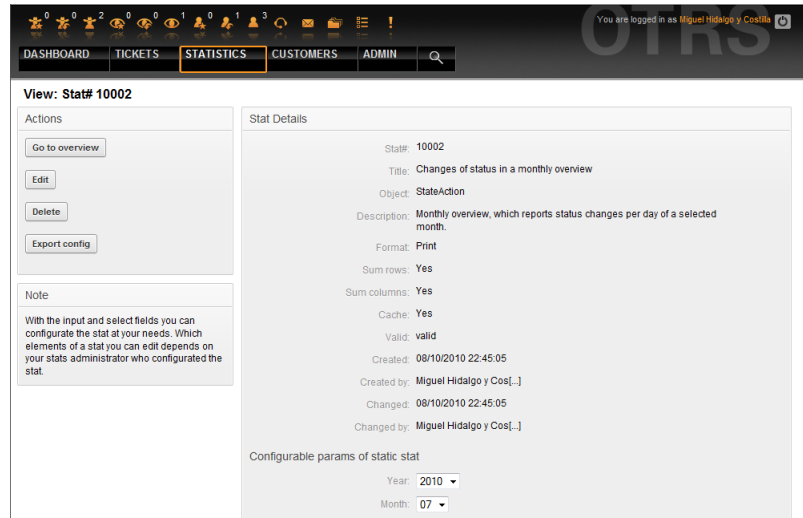


Abb.: Ansicht der statischen Statistik.

- **Anzeige dynamischer Statistiken** Dynamische Statistiken können auf zweierlei Weise angezeigt werden:
 - **Unveränderbare Einstellungen.** In diesem Fall hat der Ersteller der Statistik dem Agent, der eine Statistik generieren will, keinen Freiraum für Veränderungen gelassen.
 - **Veränderbare Einstellungen.** Bei derartigen Statistiken ist es dem Agent noch erlaubt die vorkonfigurierte Statistik zu verändern.

Um eine Statistik zu generieren, muss abschließend lediglich der "Start"-Button am rechten unteren Ende des Blocks gedrückt werden. Sollte einmal der "Start"-Button nicht sichtbar sein, kann dies zwei Ursachen haben:

1. die Statistik wurde auf ungültig gesetzt und damit deaktiviert.
2. die Statistik wurde nicht sauber konfiguriert und ist noch nicht lauffähig. Ist dies der Fall, findet man die nötigen Informationen im Notification-Bereich von OTRS (unterhalb der Navigationsleiste).

Bei falschen Einstellungen auf der Ansichtseite wird nach Drücken des "Start"-Button wieder die Ansichtseite gezeigt und im Notificationbereich darauf hingewiesen, welche Eingaben falsch waren.

Bearbeiten / Neu

Agenten mit Schreibrechten können existierende Report-Konfigurationen bearbeiten, indem sie die Bearbeitungsmaske des Statistikmoduls aufrufen. Alternativ können sie einen neuen Report erstellen. Die zugehörigen Masken können folgendermaßen aufgerufen werden:

1. Durch den Knopf "Bearbeiten" in der Statistikansicht.
2. Durch den "Neu"-Link im Statistikmenü der Navigationsleiste, oder durch den Knopf "Hinzufügen" in der Übersichtsseite.

Das Bearbeiten der Statistiken erfolgt in vier Schritten mit Hilfe eines Assistenten bzw. Wizards:

1. Allgemeine Angaben zur Statistik.
2. Festlegen des Elements für die X-Achse.
3. Festlegen der Wertereihen.
4. Festlegen der Einschränkungen des Reports.

Die Punkte 2-4 werden nur für dynamisch zusammengestellte Statistiken benötigt. Benutzt man eine statische Statistik ist man mit dem Einfügen der allgemeinen Angaben in Punkt 1. bereits fertig.

Für alle Eingabeoberflächen des Statistik-Moduls gilt: Im Anschluss an das jeweilige Eingabeformular findet der Benutzer Informationen zur Bedienung der Seite.

Sollte es dennoch zu fehlerhaften Eingaben kommen, wird die zuvor bearbeitete Oberfläche wieder geladen. Dabei wird die Seite um Informationen zur fehlerhaften Eingabe ergänzt. Diese Informationen findet man im Notificationbereich von OTRS. Erst nachdem ein Formular korrekt ausgefüllt wurde, gelangt man zur nächsten Eingabeoberfläche.

1. *Allgemeine Angaben.* Dies ist die erste Seite des Assistenten.

Abb.: Allgemeine Angaben zur Statistik bearbeiten.

Folgende allgemeine Angaben und Einstellungen der Statistik können geändert werden:

- *Titel.* Er sollte kurz und prägnant die Aufgabe der Statistik wiedergeben.
- *Beschreibung.* Hier werden Informationen eingetragen, die einen tieferen Einblick in die Aufgabenstellung, Art der Konfigurationsparameter, usw. erlauben.
- *Dynamisches Objekt.* Sollte die OTRS-Installation über mehrere dynamische Objekte verfügen, kann hier das gewünschte Objekt ausgewählt werden. Die Objekte sind jeweils auf die Anforderungen des entsprechenden Moduls ausgelegt.
- *Statische Datei.* An dieser Stelle kann eine gewünschte statische Datei ausgewählt werden. Diese Auswahl wird für gewöhnlich nicht erscheinen, da nur die statischen Dateien angezeigt werden, die noch keiner Statistik zugeordnet wurden! Wenn der Punkt "Statische Datei" angezeigt wird, ist es wichtig im Formular durch die Auswahl des Optionsfelds die gewünschte Generierungsart auszuwählen (Dynamisch durch ein

dynamisches Objekt oder statisch durch eine Datei). Wird eine statische Datei ausgewählt entfallen die Eingabeoberflächen 2-4, da die statische Datei die nötigen Konfigurationen bereits mitbringt.

- *Rechtevergabe.* Durch die Rechtevergabe kann man regeln, welche Gruppen (und dadurch welche Agents) die vorkonfigurierten Statistiken später ansehen und generieren können. So können die verschiedenen Statistiken auf die verschiedenen Abteilungen und Arbeitsgruppen aufgeteilt werden, die diese benötigen. Eine Mehrfachauswahl ist dabei möglich.

Beispiel 1: Die Gruppe "stats" wurde ausgewählt. Der Report kann von allen Nutzern verwendet werden, die mindestens "ro"-Rechte für die Gruppe "stats" haben. Diese Berechtigung ist die Voreinstellung.

Beispiel 2: Es wurde eine Gruppe namens "sales" selektiert. Alle Benutzer die ro-Rechte in der Gruppe "sales" haben können nun diese Statistik im Ansichtsmodus sehen und generieren. Anderen Benutzern, die in der Lage sind Statistiken zu generieren wird dann diese Statistik nicht angeboten.

- *Format.* Ausgabeformat der Statistik. Hierbei stehen je nach Konfiguration folgende Ausgabeformate zur Verfügung:
 - CSV.
 - Drucken.
 - Liniendiagramm.
 - Balkendiagramm.
 - Balkendiagramm (horizontal).
 - Punktdiagramm.
 - Linienpunktdiagramm.
 - Flächendiagramm.
 - Tortendiagramm.
- *Grafikgröße.* Hier kann ausgewählt werden, in welchen Größen (Pixel) die Diagramme ausgegeben werden dürfen. Diese Auswahl ist nur sinnvoll, wenn unter dem Punkt "Format" ein grafisches Ausgabeformat ausgewählt wurde. Alle generell verwendbaren Grafikgrößen werden durch den OTRS-Admin in der SysConfig definiert. Bei der Konfiguration der Statistiken kann der Ersteller alle relevanten Formate vorselektieren.
- *Zeilensummierung.* Wird im "Ja" ausgewählt, wird die Statistik durch eine Spalte ergänzt, deren Zellen die Summe der jeweiligen Zeile darstellt.
- *Spaltensummierung.* Angabe, ob die Statistik durch eine Zeile ergänzt wird, deren Zellen die Summe der jeweiligen Spalte beinhaltet.
- *Cache.* Wird im "Ja" ausgewählt, werden die errechneten Statistiken im Dateisystem gecacht. Dies spart Rechenleistung und Zeit, sollte genau diese Statistik wieder aufgerufen werden. Diese Funktion ist jedoch nur sinnvoll, wenn sichergestellt ist, dass sich der Inhalt

der Statistik nicht mehr ändert. Dies ist genau zu prüfen, bevor man die Cachingfunktion nutzt.

Caching wird automatisch verhindert, wenn die Statistik keinerlei Zeitangaben enthält (es kommen laufend neue Daten hinzu) oder wenn eine Zeitangabe in der Zukunft liegt.

Sollte eine Statistik zwischenzeitlich wieder bearbeitet werden, werden alle gecachten Daten wieder gelöscht.

- *Gültig.* Ist eine vorkonfigurierte Statistik erst im Entstehen oder darf sie aus bestimmten Gründen temporär nicht mehr ausgeführt werden so wird das Element "Gültig" auf "ungültig" gesetzt. Ferner wird dadurch die Schaltfläche "Start" rechts unten am Ende des Block ausgeblendet. Ein Generieren der Statistik ist somit nicht möglich.

2. *Definition des Elements für die X-Achse.* Hier wird eingestellt welches Element für die Darstellung der X-Achse verwendet wird, bzw. bei Tabellen, welche Spaltenbezeichnung die Statistik bekommt.

Abb.: Definition des Elements für die X-Achse.

Als erstes wählt man ein Element per Optionsfeld aus. Anschließend müssen zwei oder mehr Attribute des Elements selektiert werden. Werden keine Attribute des Elements ausgewählt, werden alle Attribute verwendet, auch solche die nach der Konfiguration der Statistik hinzukommen.

Entfernt man den Haken "Fixiert" in der Checkbox, kann der Agent, der die Statistik erstellt, die Attribute des entsprechenden Elements in der "Ansehen"-Oberfläche verändern.

Eine Besonderheit stellen Zeitelemente dar, hier ist die Zeitspanne und die Skalierung anzugeben. Die Art und Anzahl der Elemente ergibt sich durch das verwendete dynamische Objekt und ist von Objekt zu Objekt verschieden.

Wurden alle Eingaben richtig getätigt, gelangt man nach der Auswahl des "Weiter" Schalters zu dem Formular "Wertereihen". Ebenfalls gibt es die Möglichkeit, einen Bearbeitungsschritt zurück zu gehen und nochmals die Oberfläche "Allgemeine Angaben" zu bearbeiten.

3. *Wertereihen.*

Im dritten Schritt der Vorkonfiguration einer Statistik werden die Wertereihen der Statistik festgelegt. Sie bilden später die einzelnen Grafen des Diagramms (grafische Darstellung) oder die einzelnen Reihen (tabellarische Darstellung).

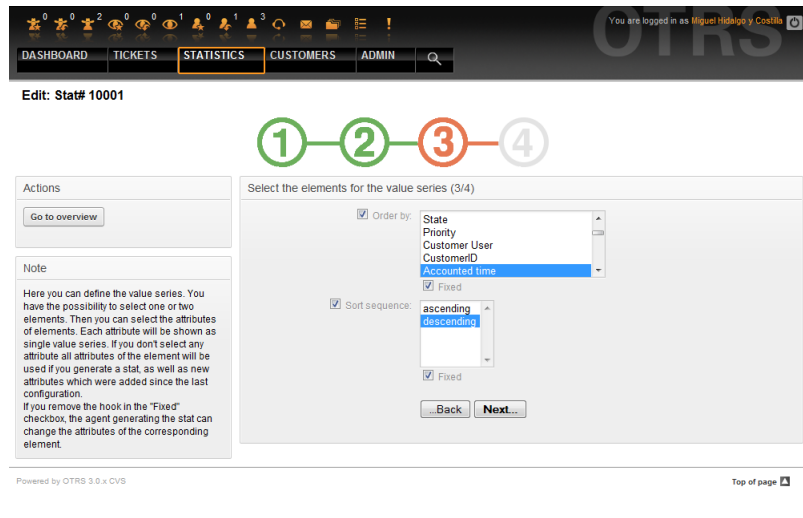


Abb.: Festlegen der Wertereihen.

Wird ein Element selektiert, entspricht jedes ausgewählte Attribut einer Wertereihe.

Beispiel 19.1. Festlegen einer Wertereihe bei einem Element

Element Queue:

- Wertereihe 1 = Raw
- Wertereihe 2 = Junk
-

Werden zwei Elemente selektiert, wird jedes ausgewählte Attribut des ersten Elements kombiniert mit einem Attribut des zweiten Elements zu einer Wertereihe.

Beispiel 19.2. Festlegen einer Wertereihe bei zwei Elementen

Element 1 Queue, Element 2 Status:

- Wertereihe 1 = Raw - offen
- Wertereihe 2 = Raw - erfolgreich geschlossen
- Wertereihe 3 = Junk - offen
- Wertereihe 4 = Junk - erfolgreich geschlossen

Die Auswahl von drei oder mehr Elementen ist nicht sinnvoll und wird deshalb durch eine Fehlermeldung verhindert.

Ferner gelten noch die gleichen Bedingungen zur Auswahl der Attribute und zur "Fixiert" Checkbox wie bei der "X-Achsen" Auswahl.

- Werden keine Attribute des Elements ausgewählt, werden alle Attribute verwendet, auch solche die nach der Konfiguration der Statistik hinzukommen.
- Entfernt man den Haken "Fixiert" in der Checkbox, kann der Agent der die Statistik erstellt, die Attribute des entsprechenden Elements verändern.

4. *Festlegen der Einschränkungen.* Vierter und letzter Schritt bei der Vorkonfiguration ist das Festlegen der Einschränkungen. Hier ist es möglich die Ergebnisse einer Statistik durch die Angabe von Bedingungen einzuschränken. Vergleichbar ist dies mit Eingaben in eine Such-Oberfläche.

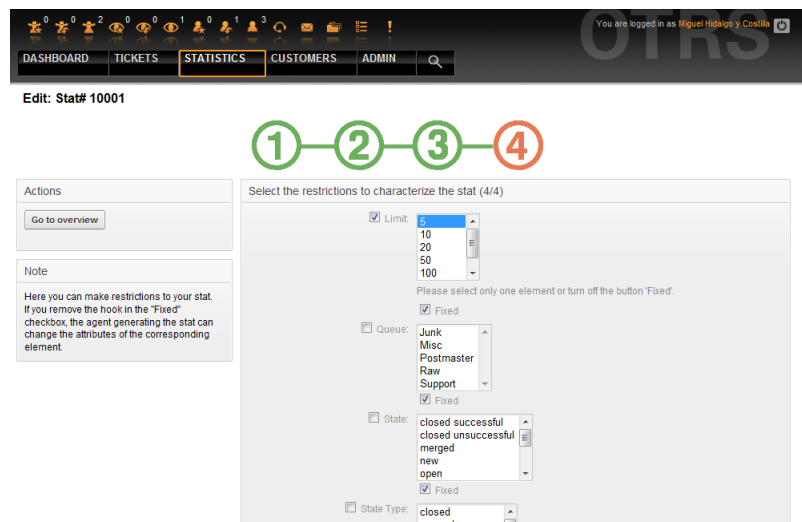


Abb.: Festlegen der Einschränkungen.

Nachdem man alle Einschränkungen getätigt hat, drückt man den Button "Abschließen". Dadurch beendet man die Vorkonfiguration der Statistik und gelangt in die "Ansehen" Oberfläche.

Importieren

Durch die Auswahl des "Import"-Menüpunkts im Statistikmenü der Navigationsleiste oder durch den "Import"-Knopf in der Übersichtsmaske gelangt man in die Importoberfläche ("rw"-Rechte erforderlich).

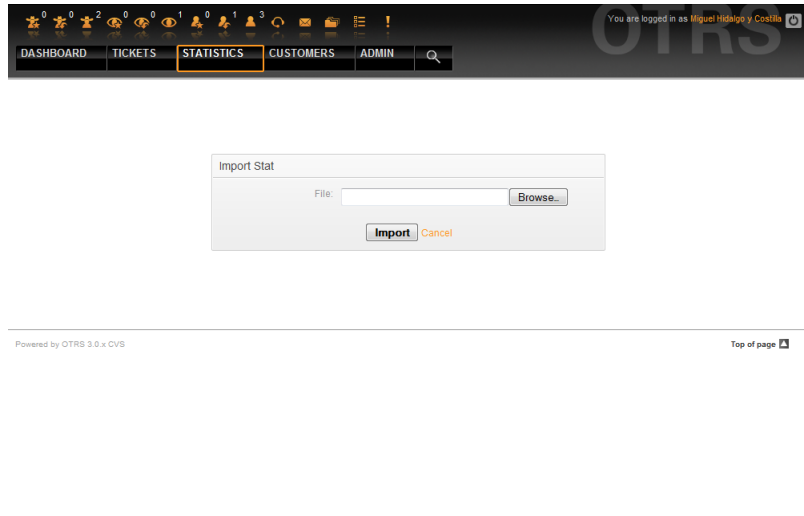


Abb.: Die Import-Oberfläche.

Durch sie können Statistiken importiert werden. Dies ist im Zusammenhang mit der Exportfunktion des Moduls eine sehr hilfreiche Funktionalität. So können Statistiken bequem auf Testsystemen entworfen und getestet werden, um sie später auf dem Produktivsystem zu importieren.

Der Import geschieht dabei ganz einfach durch einen Dateiupload. Anschließend gelangt man automatisch in die Detailansicht der importierten Statistik.

Verwaltung des Statistik-Moduls durch den OTRS-Administrator

In diesem Abschnitt werden die Aufgaben und Möglichkeiten des OTRS-Administrators in Bezug auf das Statistikmodul angesprochen.

Rechte, Gruppen und Queues

Durch die Installation des Statistikmoduls werden keine neuen Queues oder/und Gruppen angelegt.

Durch die Standardkonfiguration der Modulregistrierung haben alle Agenten, die Rechte in der "stats"-Gruppe haben, Zugriff auf das Statistikmodul.

Zugangsmöglichkeiten aufgrund der Rechte:

- *rw*. Darf Statistiken konfigurieren.
- *ro*. Darf vorkonfigurierte Statistiken generieren.

Ob nun dem jeweiligen Agenten, der später vorkonfigurierte Statistiken generieren darf, *ro*-Rechte in der "stats" Gruppe zugewiesen werden oder in der Modulregistrierung der SysConfig die dem Agent entsprechenden Gruppen ergänzt werden, bleibt dem OTRS-Administrator und dessen Anforderungen überlassen.

Die SysConfig (Verwaltung der Systemkonfiguration)

Die SysConfig-Gruppen `Framework:Core::Stats`, `Framework:Core::Stats::Graph` und `Framework:Frontend::Agent::Stats` enthalten alle Konfigurationsparameter zur grundlegenden Einstellung des Statistik-Moduls. Weiterhin regelt der Konfigurationsparameter `$Self->{'Frontend::Module'}->{'AgentStats'}` die Anordnung und Registrierung der Module und Icons innerhalb des Statistik-Moduls selbst.

Verwaltung des Statistik-Moduls durch den System-Administrator

Grundsätzlich ist bei der Bedienung, Konfiguration und Wartung des Statistikmoduls kein System-Administrator notwendig. An dieser Stelle sollen nur einige Hintergrundinformationen für den System-Administrator gegeben werden.

Anmerkung

Pfadangaben beziehen sich auf Unterverzeichnisse des OTRS-Homeverzeichnisses (meist `/opt/otrs`).

Datenbanktabelle

Alle Statistikkonfigurationen werden in XML umgesetzt und verwaltet. Dies bedeutet unter anderem, dass alle Statistikkonfigurationen in der Datenbanktabelle "xml_storage" gespeichert werden. Diese Tabelle wird auch von anderen Modulen benutzt, deren Inhalte im XML-Format dargestellt werden.

Auflistung aller Dateien

Folgende Dateien sind für die fehlerfreie Funktion des Statistikmoduls notwendig.

- `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`

Caching

Bei der Vorkonfiguration von Statistiken kann eingestellt werden, ob die Ergebnisse einer Statistik gecached werden oder nicht. Statistikergebnisse, die in den Cache geschrieben werden sollen, werden als Datei im Verzeichnis `var/tmp` der OTRS-Installation gespeichert (meist `/opt/otrs/var/tmp`).

Alle gecachten Statistiken sind dabei durch das Präfix "Stats" zu erkennen.

Sollten diese Dateien verloren gehen, ist dies nicht weiter schlimm. Beim nächsten Aufruf dieser Statistik kann das Statistikmodul diese Datei nicht mehr finden und muss deshalb die Statistik neu errechnen, was etwas Zeit beanspruchen kann.

otrs.GenerateStats.pl

Diese Datei befindet sich im Verzeichnis `bin/`. Durch sie ist es möglich, Statistiken auf der Kommandozeile zu generieren.

Exemplarisch soll hierzu folgender Kommandozeilenaufruf angegeben werden:

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

Skript: Generieren einer Statistik auf der Kommandozeile.

Dabei wird eine Statistik aus der Statistikkonfiguration "Stat# 10004" generiert und als csv in das Verzeichnis `/output/dir` gespeichert.

Ferner kann die generierte Statistik auch per Email versandt werden. Mehr Informationen erhält man wie im Skript geschildert.

```
bin> perl otrs.GenerateStats.pl --help
```

Skript: Aufruf der Hilfe von otrs.GenerateStats.pl.

Automatisiertes Generieren von Statistiken - Cron

Sinnvoll ist das Generieren von Statistiken per Hand über die Kommandozeile natürlich nicht unbedingt, da das Statistikmodul eine geeignete grafische Oberfläche bietet. Interessant wird die Generierung von Statistiken per Kommandozeile aber in Kombination mit einem Cronjob.

Folgendes Szenario ist denkbar: Ein Abteilungsleiter wünscht pünktlich zum ersten eines Monats eine Statistik zum abgelaufenen Monat. Durch die Kombination Cronjob und Kommandozeilenaufruf ist man in der Lage automatisiert zu Beginn eines Monats dem Abteilungsleiter die gewünschte Statistik per Email zukommen zu lassen.

Statische Statistiken

Das Statistikmodul bietet die Möglichkeit statische Statistiken zu generieren. Für jede statische Statistik gibt es eine Datei, in der die Inhalte der Statistik genau definiert sind.

Vorteil dieser Art von Statistik ist, dass dadurch sehr komplexe Statistiken erzeugt werden können. Der Nachteil ist wiederum die mangelnde Flexibilität dieser Statistiken.

Gespeichert werden diese Dateien unter folgendem Pfad: `Kernel/System/Stats/Static/`.

Wiederverwerten alter statischer Statistiken

Bereits OTRS 1.3 und 2.0 boten die Möglichkeit Statistiken zu generieren. Eine Vielzahl der Statistiken, die für OTRS-Versionen 1.3 und 2.0 speziell auf Kundenwunsch entwickelt wurden, können wiederverwendet werden.

Diese Dateien müssen hierzu lediglich aus dem Pfad `Kernel/System/Stats/` in den Pfad `Kernel/System/Stats/Static/` verschoben werden. Ferner muss die Paketangabe des jeweiligen Skripts um `::Static` ergänzt werden.

Das folgende Beispiel zeigt die Erweiterung des Paketnamens.

```
package Kernel::System::Stats::AccountedTime;
```

```
package Kernel::System::Stats::Static::AccountedTime;
```

Standardstatistiken

"Man muss das Rad nicht immer neu erfinden."

Deshalb liefert das Statistikmodul automatisch Standardstatistiken mit. Statistiken, die für alle OTRS-Benutzer interessant sind, werden zukünftig als Defaultstatistik im Statistik-Modul-Paket ergänzt. Gespeichert werden die Defaultstatistiken im XML-Format des Statistikmoduls im Verzeichnis `scripts/test/sample/`.

Kapitel 20. 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 in two different directions:

- *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.

Generic Interface Layers

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.

Anmerkung

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.

The list of provided Generic Interface modules shipped with OTRS will be updated and increased over time.

Figure: The graphical interface layers.

Network Transport

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 caught 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 daemon designed to process tasks asynchronously).

Data Mapping

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.

Controller

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.

Operation (OTRS as a provider)

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.

Invoker (OTRS as a requester)

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.

Generic Interface Communication Flow

Generic Interface has a defined flow to perform actions as a provider and as a requester.

This flows are described below:

OTRS as Provider

Remote Request:

1. HTTP request
 - OTRS receives HTTP request and pass it through the layers.
 - The provider module in in charge to execute and control this actions.
2. Network Transport

- The network transport module decodes the data payload and separates the operation name from the rest of the data.
- The operation name and the operation data are returned to the provider.

3. *Data External*

- Data as sent from the remote system (This is not a module-based layer).

4. 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 request data).
- The already transformed data is returned to the provider.

5. *Data Internal*

- Data as transformed and prepared to be passed to the operation (This is not a module based layer).

6. Operation

- Receives and validates data.
- Performs user access control.
- Executes the action.

OTRS Response:

1. Operation

- Returns result data to the provider.

2. *Data Internal*

- Data as returned from operation.

3. Mapping

- The data is transformed back to the Remote system format as specified in the mapping configuration (Mapping for outgoing response data).
- The already transformed data is returned to the provider.

4. *Data external*

- Data as transformed and prepared to be passed to Network Transport as response.

5. Network Transport

- Receives the data already in the Remote System format.
- Constructs a valid response for this network transport type.

6. HTTP response

- The response is sent back to the web service client.
- In the case of an error, an error response is sent to the remote system (e.g. SOAP fault, HTTP error, etc).

OTRS as Requester

OTRS Request:

1. Event Trigger Handler

- Based on the web service configuration determines if the request will be synchronous or asynchronous.
 - Synchronous
 - A direct call to the Requester is made in order to create a new request and pass it through the layers.
 - Asynchronous
 - 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 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 pass it through the layers.

2. Invoker

- Receives data from the event.
- Validates received data (if needed).
- Call core modules to complement the data (if needed).
- Return the request data structure or send a Stop Communication signal to the requester, to gracefully cancel the request.

3. *Data Internal*

- Data as passed from the invoker (This is not a module based layer).

4. Mapping

- The data is transformed to the Remote system format as specified in the mapping configuration (Mapping for outgoing response data).
- The already transformed data is returned to the requester.

5. *Data External*

- Data as transformed and prepared for sending to the remote system.

6. Network Transport

- Receives the remote operation name and the data already transformed to the Remote System format from the requester.
- Constructs a valid request for the network transport.
- Sends the request to the remote system and waits for the response

Remote Response:

1. Network transport

- Receives the response and decodes the data payload.
- Returns the data to the requester.

2. *Data External*

- Data as received from the Remote System

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).
- The already transformed data is returned to the requester.

4. *Data Internal*

- Data as transformed and ready to be passed back to the requester.

5. Invoker

- Receives return data.
- Handles the data as needed by specifically by each Invoker (included error handling if any).
- Return the Invoker result and data to the Requester.

6. Event Handler or Scheduler

- 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.

Web-Services

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 is defined what actions the web service can perform internally (Operation), what can 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.

Web Service Graphical Interface

The web service graphical user interface (GUI) is a tool that allows to construct complex web service configurations in a friendly and nice interface. It allows to:

- Create and Delete web services.
- Import and Export configurations (in YAML file format) for existing web services.
- View, Revert and Export old configurations for existing web services in the Web Service History screen.
- Track all communication logs for each web service in the Debugger screen.

Web Service Overview

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 navigation path in a "bread crumbs" style. This navigation path is useful to know exactly in which part of the web service configuration we are, and also we can jump back to any level of the configuration at any time (this action will not save any changes).

Anmerkung

To create a new web service, press the button "Add web service", and provide the needed information.

Figure: Web services overview.

Web Service Add

The only required field in this part is the web service "Name" that needs to be unique in the system and non empty. Other fields are also necessary for the configuration like the "Debug Threshold" and "Validity" but these fields are already filled with the default value for each list.

The default value for "Debug Threshold" is "debug", under this configuration all communication logs are registered in the database, each Debug Threshold value is more restrictive and discard communication logs set for lower values.

Debug Threshold levels (from lower to upper)

- Debug
- Info

- Notice
- Error

It is also possible to define the network transport protocol for "OTRS as Provider" and "OTRS as requester".

Click on the "Save" button to register the new web service in the database or click "Cancel" to discard this operation. You will now be returned to the web service overview screen.

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".

Anmerkung

To change or add more details to a web service, click on the web service name in the web service overview screen.

Figure: Web services add.

Web Service Change

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:

- Clone web service.
- Export web service.
- Import web service.
- Configuration History.
- Delete web service.
- Debugger.

Anmerkung

"Configuration history" and "Debugger" will lead you to different screens.

Web Service Clone

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.

Anmerkung

Remember the name of the web service must be unique within the system.

Click on "Clone" button to create the web service clone or "Cancel" to close the dialog.

Figure: Web service clone.

Web Service Export

The "Export web service" button gives you the opportunity to dump the configuration of the current web service into a YAML file, download it and 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.

Warnung

All stored passwords in the web service configuration will be exported as plain text.

Right after clicking the "Export web service" button a save dialog of your browser will appear, just like when you click on a file download link on a web page.

Anmerkung

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.

Figure: Web services export.

Web Service Import

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.

Click "Import" button to create a new web service from a file or "Cancel" to close the dialog.

Anmerkung

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.

Figure: Web services import.

Web Service History

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.

The "Export web service configuration" behaves exactly as the "Export web service" feature in the web service change screen. For more information refer to that section.

If changes to the current web service configuration does 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.

Warnung

Remember that any passwords stored in the web service configuration will be exported as plain text.

Please be careful when you restore a configuration because this can't be undone.

Figure: Web service history.

Web Service Delete

Sometimes it is necessary to delete a web service completely. To do this you can press on the "Delete web service" button and a new dialog will appear asking for confirmation.

Click on "Delete" to confirm the removal of the web service or on "Cancel" to close the dialog.

Warnung

Delete a web service can't be undone, please be careful when deleting a web service.

Figure: Web service delete.

Web Service Debugger

The Debugger stores the log of a web service. In the debugger screen you can track all the web service communications for either provider or requester types.

When this screen is shown the request list starts to load. After the list is fully filled you can choose one of the rows (that means a communication sequence) to check its details. This details will appear in a box below.

You can narrow the communication list using the filter on the right part of the screen. You can filter by:

- Communication type (provider or requester)
- Date: before and / or after a particular date
- The remote IP Address

- A combination of all.

After filter settings are set, push the "Refresh" button and a new list will be displayed meeting your search criteria.

Anmerkung

Depending on the search criteria for the filters the new list could return no results.

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.

In the "Request details" section you can see all the details for the selected communication. Here you can track the complete flow and check for possible errors or confirm success responses.

Figure: Web service debugger.

Web Service Configuration Change

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 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 backends 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 as requester" is very similar to the previous one, but instead of "operations" you can add invokers here.

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.

Figure: Web services change.

Anmerkung

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", when the configuration is re-visited then 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 same screen to review your changes or to configure deeper settings.

"Save and Finish" will store the current configuration level in the database and it will return to the previous screen in the configuration hierarchy (to the immediate upper configuration level).

"Cancel" will discard any configuration change to the current configuration level and will return to the previous screen in the configuration hierarchy.

Web Service Provider Network Transport

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.

For "HTTP::SOAP" protocol as provider the configuration is quite simple. 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 it's a field where you can specify the maximum size (in bytes) for SOAP messages that OTRS will process.

Figure: Web service provider network transport.

Web Service Operation

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 shares the same configuration dialog. But each operation can have independent configuration dialogs if needed.

Name, Description, Backend, and Mappings are fields that normally appear on every operation, other special fields can appear in non default configuration dialogs like the Remote System GUID field in SolMan Controller operations.

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 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-filled 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 its configuration or "Cancel" to close the delete dialog.

Figure: Web service operation.

Web Service Provider 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:

- Encoding (such as utf-8, latin1, iso-8859-1, cp1250, etc) for the SOAP message.
- 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).
- Authentication: to set the authentication mechanism, set to "-" to not use any authentication or select one from the list and the detail fields will appear.

Anmerkung

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.

Warnung

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.

Figure: Web service requester network transport.

Web Service Invoker

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), normally invokers from the same controller need similar settings and share the same configuration dialogs. Each invoker can have independent configuration dialogs if needed.

Name, Description, Backend, and Mappings are fields that normally appear on every invoker, as well as the list of event triggers other special fields can appear on non default configuration dialogs like the Remote System GUID field in SolMan Controller invokers.

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-filled and is not editable. You will see this parameter when you choose the invoker on the web service edit screen. The field is only informative. 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.

Anmerkung

The OTRS Scheduler is a separated 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 then 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 ask 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.

Figure: Web service invoker.

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 has to be changed, and sometimes even the names of the values (keys) or sometimes the complete structure, in order to match with the expected data on the other end. To accomplish this task the 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.

Anmerkung

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.

This module gives you the opportunity to set default values to map for each key or value for the whole communication data.

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:

- Keep (leave unchanged): doesn't touch the keys or values in any way.
- 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 it 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:

- Exact value(s): the old key string will be changed to a new one if the old key matches exactly.
- Regular expression: The key string will be replaced following a regular expression rule.

Pressing the new value map "+" button will display a new row for a value map. Here also is 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.

Deleting the complete key mapping section (box) is possible, just push on the "-" button located on the up right corner of each box that you want to delete.

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.

Figure: Web service mapping.

Web Service Command Line Interface

The Command Line Interface (CLI) is fast way to work with the web services. It consists of a set of tools can be use to perform basic operations like:

- Create, Update, Read, List and Delete web services based on YAML files.
- Read the Debugger log, with filter options.

Anmerkung

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.

Web Service Configuration

The "WebserviceConfig.pl" was developed in order to create basic, but fast and powerful tool to work with web service configurations. It gives you the ability to perform the following actions:

- Add: to create web services using a YAML file as the configuration source.
- Update: to change an existing web service, the configuration can be changed using a different or modified YAML file.
- Read: to get the current web service configuration displayed on the screen.
- List: to get a complete list of all the web services registered in system.
- Delete: to delete a web service from the system. Be careful when you use it, because this action can't be undone.

Warnung

A web service READ operation will display all the configuration as plain text on the screen, including any stored passwords. Please be aware of this and take the needed precautions!

Example: Creating a new web service configuration:

```
shell> OTRS_HOME/bin/otrs.WebserviceConfig.pl -a write  
-n <webservice_name> -f /path/to/yaml/file
```

Also you can use 'otrs.WebserviceConfig.pl' with following options:

- **-a read -i <webservice_id>** - To read a stored configuration.
- **-a write -n <webservice_name> -f /path/to/yaml/file** - To create a new web service.
- **-a write -i <webservice_id> -f /path/to/yaml/file** - To update a web service.
- **-a list** - To list available web services.
- **-a delete -i <webservice_id>** - To delete a web service.

Web Service Debugger

Another available tool on the command line is the "otrs.GenericInterfaceDebugRead.pl" script, which is an interface to search for web service debugger log entries.

Example: Searching for debugger log entries:

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

Optional parameters can be used for the "otrs.GenericInterfaceDebugRead.pl" script:

- **-c** - to filter by Communication ID (md5sum format).

- **-t** - to filter by CommunicationType ('Provider' or 'Requester').
- **-a** - to filter by date (At or After a date).
- **-b** - to filter by date (At or Before a date).
- **-i** - to filter by IP Address (must be valid IPv4 or IPv6 address).
- **-w** - to filter by Web Service ID.
- **-d** - to include detailed communication data.

Example: Searching for debugger log entries with all parameters:

```
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
```

Anmerkung

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.

Web Service Configuration

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 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.

Anmerkung

To learn more about YAML please visit <http://www.yaml.org/>.

The following is a web service configuration file example in YAML format:

```
---
Debugger:
  DebugThreshold: debug
Description: This an example of a web service configuration
Provider:
  Operation:
    CloseIncident:
      Description: This is a SolMan test operation
```

```
MappingInbound: {}
MappingOutbound: {}
RemoteSystemGuid: ''
Type: SolMan::CloseIncident
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
```

Configuration Details

General

- Description: a short text that describes the web service.
- RemoteSystem: a short description of the Remote System.
- Debugger: a container for the debugger settings.
- Provider: a container for the provider settings.
- Requester: a container for the requester settings.

Debugger

- DebugThreshold: the debugger level

Possible Values

- debug: all logs are stored in the database.
- info: info, notice and error level logs are stored in the database.
- notice: notice and error level logs are stored in the database.
- error: only error level logs are stored in the database.

Provider

- Operation: a container for each operation settings.
- Transport: a container for provider network transport settings.

Operation

- <OperationName>: Unique name for the operation, container for its own operation settings (cardinality 0..n, but not duplicate).

<OperationName>

This section is based on operations from type "Test::Test" other operations might contain more or different settings.

- Description: a short text that describes the operation.
- MappingInbound: a container for the mapping settings for the incoming request data.
- MappingOutbound: a container for the mapping settings for the outgoing response data.
- Type: the operation backend, in Controller::Operation format.

MappingInbound

This section is based on mappings from type "Simple". Other mappings might contain more or different settings.

- Config: a container for this mapping settings.
- Type: the mapping backend.

Config

- KeyMapDefault: a container for all non mapped keys settings.
- ValueMapDefault: a container for all non mapped values settings.
- KeyMapExact: a container for all exact key mappings (cardinality 0 .. 1).
- KeyMapRegEx: a container for all regular expression key mappings (cardinality 0 .. 1).
- ValueMap: a container for all value mappings (cardinality 0 .. 1).

KeyMapDefault

- MapTo: the new value to be used (only applicable if MapType is set to MapTo).
- MapType: the rule for the mapping.

Possible Values

- Keep: leave unchanged.
- Ignore: drop.
- MapTo: change to the MapTo value.

ValueMapDefault

Similar to KeyMapDefault.

KeyMapExact

- <oldkey>: <newkey> (cardinality 0 .. n but not duplicate).

KeyMapRegEx

- <oldkey(RegEx)>: <newkey> (cardinality 0 .. n but no duplicates).

ValueMap

- <newkey>: a container for value mappings for this new key (cardinality depends on the new keys from KeyMapExact and KeyMapRegEx).

<newkey>

- ValueMapExact: a container for all exact value mappings (cardinality 0 .. 1).
- ValueMapRegEx: a container for all regular expression value mappings (cardinality 0 .. 1).

valueMapExact

- <oldvalue>: <newvalue> (cardinality 0 .. n but not duplicate).

ValueMapRegEx

- <oldvalue(RegEx)>: <newvalue> (cardinality 0 .. n but not duplicate).

MappingOutbound

Same as MappingInbound.

Transport

This section is based on the provider network transport HTTP::SOAP, other transports might contain more or different settings.

- Config: a container for the specific network transport configuration settings.
- Type: the provider network transport backend.

Config

- MaxLength: the maximum length in bytes to be read in a SOAP message by OTRS.
- NameSpace: an URI that gives a context to all operations that belongs to this web service.

Requester

- Invoker: a container for each invokers' settings.
- Transport: a container for requester network transport settings.

Invoker

- <InvokerName>: Unique name for the invoker, container for its own invoker settings (cardinality 0..n, but not duplicate).

<InvokerName>

This section is based on invokers from type "Test::Test" other invokers might contain more or different settings.

- Description: a short text that describes the invoker

- Events: a container for a unnamed list of event trigger settings.
- MappingInbound: a container for the mapping settings for the incoming response data.
- MappingOutbound: a container for the mapping settings for the outgoing request data.
- Type: the invoker backend, in Controller::Invoker format.

Events

- *List Element*: (cardinality 0 .. n)
- Asynchronous: to set if the invoker execution will be delegated to the Scheduler

Possible Values

- 0: not handled by the Scheduler.
- 1: handled by the Scheduler.
- Event: the name of the event trigger.

Possible Values (for ticket events)

- 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

Possible Values (for article events)

- Article Events
- ArticleCreate
- ArticleFreeTextUpdate
- ArticleUpdate
- ArticleSend
- ArticleBounce
- ArticleAgentNotification
- ArticleCustomerNotification
- ArticleAutoResponse
- ArticleFlagSet
- ArticleFlagDelete
- ArticleAgentNotification
- ArticleCustomerNotification

MappingInbound

Same as Operation MappingInbound

MappingOutbound

Same as Operation MappingInbound.

Transport

This section is based on the requester network transport HTTP::SOAP, other transports might contain more or different settings.

- Config: a container for the specific network transport configuration settings.
- Type: the requester network transport backend.

Config

- Authentication: a container for authentication settings.
- Encoding: the SOAP Message request encoding
- Endpoint: the URI of the Remote Server web service to accept OTRS requests
- NameSpace: an URI that gives a context to all invokers that belongs to this web service.
- SOAPAction: to send an empty or filled SOAPAction header in the SOAP Message (in "<NameSpace> <Separator> <Action>" format).

Possible Values

- YES: to send a filled SOAPAction header.
- No: to send an empty SOAPAction header.
- SOAPActionSeparator: to set the <Separator> of a filled SOAPAction header.

Possible Values

- '/': used for .net web services.
- '#': used for all the rest web services.

Authentication

- User: the privileged user name that has access to the remote web service.
- Password: the password for privileged user in plain text.
- Type: the type of authentication.

Connectors

A Connector is in essence set of actions 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 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 so specific for the Connector that implements them.

In another 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.

Bundled Connectors

Included with this version of OTRS the following connectors are ready to be used.

- Session
- Ticket

Session Connector

This connector is capable to create a valid SessionID that can be used in any other operation.

Provides:

- Operations:
 - SessionCreate

Operations

SessionCreate

Creates a new new valid SessionID to be used in other operations from other connectors like TicketCreate.

Anmerkung

To use the SessionID in other operations from other connectors is necessary that the operation implements authentication by SessionID. all the rest of the bundled operations are capable to accept a valid SessionID as an authentication method.

Possible Attributes:

```
<SessionCreate>
  <!--You have a MANDATORY CHOICE of the next 2 items at this
level-->
  <!--Optional:-->
  <UserLogin>?</UserLogin>
  <!--Optional:-->
  <CustomerUserLogin>?</CustomerUserLogin>
  <!--Optional:-->
  <Password>?</Password>
</SessionCreate>
```

Ticket Connector

This connector supplies the basic functionality to interact with tickets

Provides:

- Operations:
 - TicketCreate
 - TicketUpdate
 - TicketGet
 - TicketSearch

Operations

TicketCreate

Provides an interface to create 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.

Possible Attributes:

```
<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:-->
        <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>

```

TicketUpdate

TicketUpdate operation add the capability to modify attributes from a ticket or add a new article, including attachments and all defined dynamic fields for the ticket and the new article.

Anmerkung

It is not necessary to create a new article to modify a ticket attribute.

Possible Attributes:

```

<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:-->
<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>
```

TicketGet

This operation is used to get all the attributes of a ticket including the dynamic fields, all the articles and all the attachments that belongs to the ticket.

Possible Attributes:

```
<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>
  <!--Optional:-->
  <TicketID>?</TicketID>
  <!--Optional:-->
  <DynamicFields>?</DynamicFields>
  <!--Optional:-->
  <Extended>?</Extended>
  <!--Optional:-->
  <AllArticles>?</AllArticles>
  <!--Optional:-->
  <DynamicFields>?</DynamicFields>
  <!--Optional:-->
  <ArticleSenderType>?</ArticleSenderType>
  <!--Optional:-->
  <ArticleOrder>?</ArticleOrder>
  <!--Optional:-->
  <ArticleLimit>?</ArticleLimit>
  <!--Optional:-->
  <Attachments>?</Attachments>
</TicketGet>
```

TicketSearch

TicketSearch operation returns a list of Ticket IDs that matches a predefined criteria.

Possible Attributes:

```
<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:-->
  <TypeID>?</TypeID>
  <!--Zero or more repetitions:-->
  <States>?</States>
  <!--Zero or more repetitions:-->
  <StateIDs>?</StateIDs>
  <!--Zero or more repetitions:-->
  <StateType>?</StateType>
  <!--Zero or more repetitions:-->
  <StateTpeyIDs>?</StateTpeyIDs>
  <!--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:-->
  <SLA>?</SLA>
  <!--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:-->
<CreatedTypeID>?</CreatedTypeID>
<!--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:-->
<CreatedStateIDs>?</CreatedStateIDs>
<!--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:-->
  <LowerThan>?</LowerThan>
  <!--Optional:-->
  <LowerThanEquals>?</LowerThanEquals>
</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:-->
<ContentSearchPrefix?></ContentSearchPrefix>
<!--Optional:-->
<ContentSearchSuffix?></ContentSearchSuffix>
<!--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>
<!--Optional:-->
<Permission>?</Permission>
<!--Zero or more repetitions:-->
<CustomerUserID>?</CustomerUserID>
</TicketSearch>
```

Examples:

Web Service Configuration

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: 3.1.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: ''
```

Perl SOAP Requester

The following code is a Perl script that can connect to OTRS via the generic interface, to perform the operations provided by the Ticket Connector, it uses two Perl CPAN modules SOAP::Lite and Data::Dumper, be sure that your environment is capable to use that modules before you try to run the script.

```
#!/usr/bin/perl -w
# --
# otrs.SOAPRequest.pl - sample to send a SOAP request to OTRS Generic
# Interface Ticket Connector
# Copyright (C) 2001-2012 OTRS AG, http://otrs.org/
# --
# $Id: genericinterface-connectors.xml,v 1.1 2012/02/23 13:44:35 mg
# Exp $
# --
# 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 OTRS Admin Manual in order to get the complete list.
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} );  
}  
}
```

Kapitel 21. OTRS Scheduler

The OTRS Scheduler is an independent system process that executes tasks in background. These kind of processes are known 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.

Currently the OTRS Scheduler is only able to handle Generic Interface tasks. These kind of tasks executes invokers that send requests to remote systems. Other handlers for different tasks will be added in future OTRS versions.

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 set it and forget it process, the only needed human interaction is to check its status periodically and start or stop it as needed.

Anmerkung

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).

Scheduler Graphical Interface

The Scheduler is not visible in the OTRS Graphical User Interface unless it stops running.

Scheduler Not Running Notification

There are two different types of notifications if the system detects that scheduler is not running. This detection is based on the Scheduler process update frequency, 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.

The Scheduler process update time can be configured via the SysConfig in the "Scheduler::PIDUpdateTime" setting.

If you would see a warning message it is not always necessary to take an action, but is highly recommended to check if the scheduler process is running. If you see an alert, then is highly probable that the scheduler is in fact not running and should be started.

By default the Scheduler not running notification is enabled, if there is a valid web service registered in the database, and is only displayed to the users in the "admin" group.

To disable the notification (not recommended) or to change or add the notification groups, please edit the "Frontend::NotifyModule###800-Scheduler-Check" setting in the SysConfig.

Figure: Scheduler notification.

Start Scheduler

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.

Anmerkung

A forced Scheduler start is only necessary if previous Scheduler process was terminated abnormally and the Process ID is still registered in the database.

To have full control of the Scheduler process and to check its real status please use the command line tools described below.

Figure: Start Scheduler.

Scheduler Command Line Interface

The Scheduler command line tools let you control the Scheduler process (Start / Stop) or query its status. There are also tools to register the process to be controlled by the operating system.

Included with OTRS there are two sets of CLI tools, one for Unix / Linux OS and another for MS Windows OS.

Unix / Linux

Scheduler Init.d Files

Init.d files are special scripts that are called by the operating system at startup and shutdown (or restart) times.

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`.

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.

Init.d Script Internal Variables

- **OTRS_HOME** - the path of your OTRS installation.
- **User** - the apache process user name.
- **Group** - the apache process user's group name.

Anmerkung

Currently there are only init.d scripts for Linux platforms.

Tabelle 21.1. List of Init Scripts And Supported Operating Systems

Init Script	Supported OS
otrs-scheduler-linux	Red Hat, Fedora, CentOS, SUSE, openSUSE, Debian, Ubuntu

Init Script	Supported OS
otrs-scheduler-gentoo-init.d, otrs-scheduler-gentoo-conf.d	Gentoo

Beispiel 21.1. Example To Start The OTRS Scheduler Form An Init.d Script

```
shell> /etc/init.d/otrs-scheduler-linux  
start
```

Available Actions

- **start** to start the OTRS Scheduler process.
- **stop** to stop the OTRS Scheduler process.
- **restart** to restart the OTRS Scheduler process.
- **status** to query the OTRS Scheduler process status.

The Scheduler needs the database to be available to register its Process ID, for this reason is necessary to:

- Execute the Scheduler init.d script to *start* the Scheduler process after the database process is up and running.
- Execute the Scheduler init.d script to *stop* the Scheduler before the database process shuts down.

Anmerkung

If you want the Scheduler to run at system startup, please read the documentation of the operating system for the right location to place the init.d scripts, how to configure them to run automatically and how to set the run order.

Scheduler Daemon File

This is the part of the Scheduler that stays running in the background checking for tasks to execute. It also provides the main functions to control the process.

All Unix / Linux uses the file **OTRS_HOME/bin/otrs.Scheduler.pl**.

Beispiel 21.2. Example To Start The OTRS Scheduler

```
shell> OTRS_HOME/bin/otrs.Scheduler.pl -a  
start
```

Available Options

- **-a** action.

Possible Values

- **start**- to start the Scheduler process.
- **stop**- to stop the Scheduler process.
- **status**- to query Scheduler process status.
- **-f** to force the start or stop of the Scheduler process.

Beispiel 21.3. Example To Force Stop The OTRS Scheduler

```
shell> OTRS_HOME/bin/otrs.Scheduler.pl -a  
stop -f 1
```

Anmerkung

Force stop the Scheduler is used remove the process ID from the database when 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.

Windows

Scheduler Service Installer

The integration of the services into the MS Windows Operating System is done via the Windows Service Control Manager (SCM). In order to make the OTRS Scheduler process to be controlled by the SCM is necessary to register this service

OTRS provides the script **OTRS_HOME/bin/otrs.Scheduler4WinInstaller.pl** to register or unregister the OTRS Scheduler into the SCM.

Beispiel 21.4. Example To Register The OTRS Scheduler Into the Widows SCM

```
shell> OTRS_HOME/bin/  
otrs.Scheduler4WinInstaller.pl -a install
```

Available Options

- **-a** action.

Possible Values

- **install**- to install the Scheduler process into the Windows SCM.
- **remove**- to remove the Scheduler process from the Windows SCM.

After installing into the Windows SCM the OTRS Scheduler process can be used as any other service in Windows. It can be started, stopped and restarted and can be configured to be started manually or automatic.

Anmerkung

To learn more about Windows Services and the Windows SCM please read the Windows documentation, and Microsoft online help.

Scheduler Service File

This is the part of the Scheduler that stays running in the background checking for tasks to execute. It also provides the main functions to control the process.

Windows Operating System uses the file **OTRS_HOME/bin/otrs.Scheduler4Win.pl**.

Beispiel 21.5. Example To Start The OTRS Scheduler

```
shell> OTRS_HOME/bin/otrs.Scheduler4Win.pl  
-a start
```

Available Options

- **-a** action.

Possible Values

- **start**- to start the Scheduler process.
- **stop**- to stop the Scheduler process.
- **status**- to query Scheduler process status.
- **-f** to force the start or stop of the Scheduler process.

Beispiel 21.6. Example To Force Stop The OTRS Scheduler

```
shell> OTRS_HOME/bin/otrs.Scheduler4Win.pl  
-a stop -f 1
```

Anmerkung

Force stopping the Scheduler is used to remove the process ID from the database when 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.

Kapitel 22. Dynamic Fields

Introduction

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 were 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 built 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.

The following dynamic field types are included with this release:

- Text (one line of text)
- Textarea (multiple lines of text)
- Checkbox
- Dropdown (single choice, multiple values)
- Multiselect (multiple choice, multiple values)
- Date
- Date / Time

Configuration

By default, a clean installation of OTRS 3.1.x does not include any dynamic fields. If you plan to use such fields in tickets or articles you need to create dynamic fields.

An updated installation from OTRS 3.0.x will have all the old "free fields" created as dynamic fields for compatibility and data preservation. The following is the list of dynamic fields that are created during the migration from OTRS 3.0.x to 3.1.x.

- TicketFreeKey[1-16] (TicketFreeKey1, TicketFreeKey2 ... TicketFreeKey16)
- TicketFreeText[1-16]

- TicketFreeTime[1-6]
- ArticleFreeKey[1-16]
- ArticleFreeText[1-16]

Anmerkung

During the migration procedure from OTRS 3.0.x to OTRS 3.1.x all the old "free fields" data and configuration are migrated to the new dynamic fields architecture. Any custom development around the old "free fields" has to be updated to use the new dynamic field framework.

The migration of the configuration include the field itself and the screen configurations to hide, show or show field as mandatory for each screen.

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.

Adding a Dynamic Field

Click in 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:

Figure: Dynamic fields overview screen, empty.

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.

Anmerkung

The installation of an OTRS package could add more objects to the Action side bar.

The general procedure to create a dynamic field is:

- Click on the desired dynamic field object dropdown in the Action side bar.
- Click on the dynamic field type that you want to add from the list.
- Fill the configuration.
- Save.

The configuration dialogs for the dynamic fields are split in two parts, the upper section is common among all the fields and the lower part might be different from one type of dynamic field to another.

General dynamic field settings:

- Name: Mandatory, unique, only letters and numbers are allowed.

This is the internal name of the field, used for example to show or hide a field in one screen. Any modification on field name (not recommended) will need manual a update on the "SysConfig" settings where the field is referenciated.

- Label: Mandatory.

This is field name to be displayed in the screens, it supports translations.

Anmerkung

Label translations have to be added manually to language translations files.

- Field order: Mandatory.

Defines the relative order in which the field will be displayed in the screen, by default each new field has the last position, a change in this setting will affect the other of the other created dynamic fields.

- Validity: Mandatory.

An invalid dynamic field will not be displayed in any screen, no matter if is configured to displayed.

- Field type: Mandatory, Read only.

Shows the current selected field type.

- Object type: Mandatory, Read only.

Shows the scope of field.

Anmerkung

To illustrate each specific field type settings a few fields will be added in our example. These new fields will be referenciated in later sections.

For the following examples all the dynamic fields will be created for the Ticket object if you need to create a dynamic field for Article object, just chose the field from the Article dropdown list.

Tabelle 22.1. The following fields will be added into the system:

Name	Label	Type
Field1	My Field 1	Text
Field2	My Field 2	Textarea
Field3	My Field 3	Checkbox
Field4	My Field 4	Dropdown
Field5	My Field 5	Multiselect
Field6	My Field 5	Date
Field7	My Field 6	Date / Time

Text Dynamic Field Configuration

Text dynamic field is used to store a single line string.

Text dynamic field settings:

- Default value: Optional.

This is the value to be shown by default in the edit screens (like New Phone Ticket or Ticket Compose).

- Show link: Optional.

If set, the field value will be converted into a clickable link for display screens (like ticket zoom or overviews).

For example, if "Show link" is set to "http://www.otrs.com", clicking on the filled value will make your browser to open the OTRS web page.

Anmerkung

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.

Figure: Dynamic field Text configuration dialog.

Textarea Dynamic Field Configuration

Textarea dynamic field is used to store a multiple line string.

Textarea dynamic field settings:

- Number of rows: Optional, integer.

Used to define the height of the field in the edit screens (like New Phone Ticket or Ticket Compose).

- Number of cols: Optional, Integer.

This is value is used to define the width of the field in the edit screens.

- Default value: Optional.

This is the value to be shown by default in the edit screens (it can be a multiple line text).

Figure: Dynamic field Textarea configuration dialog.

Checkbox Dynamic Field Configuration

Checkbox dynamic field is used to store true or false value, represented by a checked or unchecked check box.

Checkbox dynamic field settings:

- Default value: Mandatory.

This is the value to be shown by default in the edit screens (like New Phone Ticket or Ticket Compose), the default value for this field is closed selection that can be Checked or Unchecked.

Figure: Dynamic field Checkbox configuration dialog.

Dropdown Dynamic Field Configuration

Dropdown dynamic field is used to store a single value, from a closed list.

Dropdown dynamic field settings:

- Possible values: Mandatory.

List of values to choose. when add a new value is necessary to specify the Key (internal value) and the Value (display value).

- Default value: Optional.

This is the value to be show by default in the edit screens (like New Phone Ticket or Ticket Compose), the default value for this field is closed selection defined by the Possible values.

- Add empty value: Mandatory, (Yes / No).

If this option is activated an extra value is defined to show a "-" in the list of possible values, this special value is empty internally.

- Translatable values: Mandatory, (Yes / No).

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.

- Show link: Optional.

If set, the field value will be converted into a clickable HTP link for display screens (like Zoom or overviews).

For example, if Show link is set to "http://www.otrs.com", clicking on the filed value will make your browser to open the OTRS web page.

Anmerkung

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.

Figure: Dynamic field Dropdown configuration dialog.

Multiselect Dynamic Field Configuration

Multiselect dynamic field is used to store a multiple values, from a closed list.

Multiselect dynamic field settings:

- Possible values: Mandatory.

List of values to choose. when add a new value is necessary to specify the Key (internal value) and the Value (display value).

- Default value: Optional.

This is the value to be show by default in the edit screens (like New Phone Ticket or Ticket Compose), the default value for this field is closed selection defined by the Possible values.

- Add empty value: Mandatory, (Yes / No).

If this option is activated an extra value is defined to show a "-" in the list of possible values, this special value is empty internally.

- Translatable values: Mandatory, (Yes / No).

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.

Figure: Dynamic field Multiselect configuration dialog.

Date Dynamic Field Configuration

Date dynamic field is used to store a date value (Day, Month and Year).

Date dynamic field settings:

- Default date difference: Optional, Integer.

Number of seconds (positive or negative) between the current date and the selected date to be show by default in the edit screens (like New Phone Ticket or Ticket Compose).

- Define years period: Mandatory (Yes / No).

Used to set a defined number of years in past and future from current date in the year select of this field, If set to Yes the following options are available:

- Years in the past: Optional, Positive integer.

Define the number of years in past from current day to display in the year selection for this field in edit screens.

- Years in the future: Optional, Positive integer.

Define the number of years in future from current day to display in the year selection for this field in edit screens.

- Show link: Optional.

If set, the field value will be converted into a clickable HTTP link for display screens (like Zoom or overviews).

For example, if Show link is set to "http://www.otrs.com", clicking on the field value will make your browser to open the OTRS web page.

Anmerkung

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.

Figure: Dynamic field Date configuration dialog.

Date / Time Dynamic Field Configuration

Date / Time dynamic field is used to store a date time value (Minute, Hour, Day, Month and Year).

Date / Time dynamic field settings:

- Default date difference: Optional, Integer.

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).

- Define years period: Mandatory (Yes / No).

Used to set a defined number of years in past and future from current date in the year selection of this field. If set to Yes the following options are available:

- Years in the past: Optional, Positive integer.

Define the number of years in past from current day to display in the year selection for this field in edit screens.

- Years in the future: Optional, Positive integer.

Define the number of years in future from current day to display in the year selection for this field in edit screens.

- Show link: Optional.

If set, the field value will be converted into a clickable HTTP link for display screens (like Zoom or overviews).

For example, if Show link is set to "http://www.otrs.com", clicking on the field value will make your browser to open the OTRS web page.

Anmerkung

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.

Figure: Dynamic field Date / Time configuration dialog.

Editing a Dynamic Field

A filled dynamic field overview screen (with the previous examples) should look like:

Figure: Dynamic field overview screen filled with sample data.

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 it's settings.

Anmerkung

Not all the dynamic field settings can be changed, the Field type and Object type are fixed from the selection of the field and they can't be changed.

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 has a reference to that particular field needs to be updated as well as user preferences (if defined).

Showing a Dynamic Field on a Screen

To display a dynamic field on a particular screen there are two mandatory conditions:

1. The dynamic field must be valid.
2. The dynamic field must be set to 1 or 2 in the configuration of the screen.

Follow this steps to show a dynamic field in a screen

- Be sure that the dynamic field is set to valid, you can see the validity of the field from the dynamic field overview screen. Set to valid by editing the field if necessary.
- Open the "sysconfig" and select "Ticket" from the dropdown list in the Actions side bar located in the left part of the screen.

Anmerkung

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 filed 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.

Show Examples

The following are "sysconfig" configurations examples to show or hide dynamic fields on different screens.