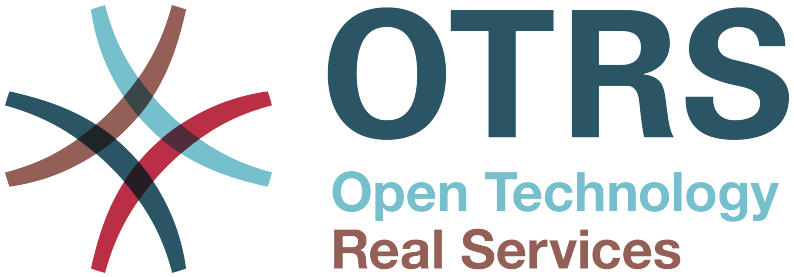


Documentation

OTRS 3.1 - Administrator-Handbuch

Build Date:

2012-02-14

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 Oschwald, 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 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
    1. Was ist ein Trouble-Ticket-System, und warum benötigen Sie eins? 1
    2. Was ist ein Trouble-Ticket? 2
2. OTRS-Helpdesk 3
    1. Grundlagen 3
    2. Features 3
        2.1. New features of OTRS 3.1 5
        2.2. Die wichtigsten neuen Features von OTRS 3.0 9
        2.3. Neue Features von OTRS 2.4 10
    3. Hardware und Software-Anforderungen 12
        3.1. Perl support 13
        3.2. Web server support 13
        3.3. Database support 13
        3.4. Web browser support 13
    4. Community 14
    5. Kommerzieller Support und Dienstleistungen für OTRS 14
3. Installation 15
    1. Der einfache Weg - Installation fertiger Pakete 15
        1.1. Installation des RPM auf einem SuSE-Linux-Server 15
        1.2. Installation von OTRS auf einem CentOS-System 16
        1.3. Installation von OTRS auf einem Debian-System 16
        1.4. Installation von OTRS auf einem Ubuntu-System 17
        1.5. Installation von OTRS auf Microsoft Windows-Systemen 17
    2. Manuelle Installation (Linux, Unix) 17
        2.1. Vorbereiten der manuellen Installation 17
        2.2. Installation der für OTRS benötigten Perl-Module 18
        2.3. Konfiguration des Apache-Webservers 20
        2.4. Einrichten der Datenbank 21
        2.5. Einrichten der von OTRS benötigten cron-Jobs 28
    3. Upgrade des OTRS Frameworks 31
    4. Upgrade mit dem Windows-Installer 36
    5. Upgrading Microsoft SQL Server Data Types 36
4. Erste Schritte 38
    1. Agenten-Weboberfläche 38
    2. Kunden-Weboberfläche 38
    3. Öffentliche Weboberfläche 39
    4. Die erste Anmeldung 40
    5. Agenten-Weboberfläche - ein Überblick 40
    6. Was verbirgt sich hinter dem Begriff Queue? 42
    7. Benutzereinstellungen 43
5. Der Administrationsbereich von OTRS 46
    1. Grundlagen 46
    2. Benutzer, Gruppen und Rollen 46
        2.1. Benutzer 46
        2.2. Gruppen 47
        2.3. Rollen 51
    3. Kundenbenutzer und Kundengruppen 55
        3.1. Kunden 55
        3.2. Kundengruppen 56
    4. Queues 58

5. Anreden, Signaturen, Anlagen und Antwortvorlagen 60
5.1. Anreden 60
5.2. Signaturen 61
5.3. Anlagen 63
5.4. Antwortvorlagen 65
6. Automatische Antworten 67
7. E-Mail-Adressen 70
8. Benachrichtigungen 71
9. S/MIME 73
10. PGP 74
11. Status 74
12. Die SysConfig (Verwaltung der Systemkonfiguration) 76
13. Einrichten von Mail-Konten 76
14. Filtering incoming email messages 77
15. Ausführen von automatisierten Jobs mit Hilfe des GenericAgents 80
16. Administrator-E-Mail 81
17. Sitzungsverwaltung 82
18. System Log 83
19. SQL-Abfragen mit Hilfe der SQL Box 83
20. Paketverwaltung 84
21. Web-Services 85
22. Dynamic Fields 86
6. Systemkonfiguration 87
1. Die Konfigurationsdateien von OTRS 87
2. Konfiguration des Systems mit Hilfe des grafischen Konfigurations-Frontends..... 87
7. E-Mails versenden/empfangen 89
1. E-Mails versenden 89
1.1. Via Sendmail (Standard) 89
1.2. Via SMTP server/smarthost 89
2. E-Mails empfangen 89
2.1. E-Mail-Konten, die über die OTRS-Oberfläche konfiguriert werden 89
2.2. Via Kommandozeilen-Programm und z. B. procmail (otrs.PostMaster.pl)..... 90
2.3. E-Mails via POP3 oder IMAP und fetchmail für otrs.PostMaster.pl empfangen 91
2.4. Filterung/Verteilung über PostMaster-Module (für komplexere Verteilungsszenarien) 91
8. Zeitabhängige Funktionen in OTRS 93
1. Arbeitszeiten, Feiertage und Zeitzonen für das System festlegen 93
1.1. Arbeitszeiten 93
1.2. Feiertage mit festem Datum 93
1.3. TimeVacationDaysOneTime 93
2. Automatische Ticket-"Freigabe" (unlock) 93
9. Ticket-Verantwortlicher und Beobachten von Tickets 95
1. Ticket-Verantwortlicher 95
2. Tickets beobachten 96
10. Customizing the PDF output 98
11. Einbinden externer Backends 99
1. Kundenbenutzerdaten 99
2. Kundenbenutzer Backend 99
2.1. Datenbank (Standard) 99
2.2. LDAP 102
2.3. Verwenden mehrerer Kunden Backends 104
3. Backends für die Authentifizierung von Agenten und Kunden 107
3.1. Authentifizierungs-Backends für Agenten 107

3.2. Authentifizierungs-Backends für Kunden ... 109
4. Customizing the customer self-registration ... 111
4.1. Anpassen der Weboberfläche ... 111
4.2. Kunden-Mapping ... 111
4.3. Customizing the customer_user table in the OTRS DB ... 112
12. Status ... 114
1. Vordefinierte Ticket-Status ... 114
1.1. Neu ... 114
1.2. Öffnen ... 114
1.3. Warten zur Erinnerung ... 114
1.4. Warten auf erfolglos schließen ... 114
1.5. Warten auf erfolgreich schließen ... 114
1.6. Zusammengefügt ... 114
1.7. Erfolgreich geschlossen ... 114
1.8. Erfolglos geschlossen ... 114
2. Anpassen von Ticket-Status ... 114
13. Anpassen der Ticket-Prioritäten ... 118
14. Erstellung eigener Themes ... 119
15. Übersetzung des OTRS-Frontends ... 120
16. PGP ... 121
17. S/MIME ... 124
18. Access Control Lists (ACLs) ... 128
1. Introduction ... 128
2. Examples ... 128
3. Reference ... 131
19. Statistikmodul ... 135
1. Bedienung des Moduls durch den Agent ... 135
1.1. Übersicht ... 135
1.2. Ansehen und Generieren von Statistiken ... 136
1.3. Bearbeiten / Neu ... 139
1.4. Importieren ... 144
2. Verwaltung des Statistik-Moduls durch den OTRS-Administrator ... 145
2.1. Rechte, Gruppen und Queues ... 145
2.2. Die SysConfig (Verwaltung der Systemkonfiguration) ... 145
3. Verwaltung des Statistik-Moduls durch den System-Administrator ... 146
3.1. Datenbanktabelle ... 146
3.2. Auflistung aller Dateien ... 146
3.3. Caching ... 146
3.4. otrs.GenerateStats.pl ... 147
3.5. Automatisiertes Generieren von Statistiken - Cron ... 147
3.6. Statische Statistiken ... 147
3.7. Wiederverwerten alter statischer Statistiken ... 147
3.8. Standardstatistiken ... 148
20. Generic Interface ... 149
1. Generic Interface Layers ... 149
1.1. Network Transport ... 150
1.2. Data Mapping ... 150
1.3. Controller ... 151
1.4. Operation (OTRS as a provider) ... 151
1.5. Invoker (OTRS as a requester) ... 151
2. Generic Interface Communication Flow ... 151
2.1. OTRS as Provider ... 151
2.2. OTRS as Requester ... 152
3. Web-Services ... 154

4. Web Service Graphical Interface ... 154
4.1. Web Service Overview ... 155
4.2. Web Service Add ... 155
4.3. Web Service Change ... 157
5. Web Service Command Line Interface ... 171
5.1. Web Service Configuration ... 171
5.2. Web Service Debugger ... 172
6. Web Service Configuration ... 173
6.1. Configuration Details ... 174
7. Connectors ... 180
7.1. Bundled Connectors ... 180
7.2. Examples: ... 188
21. OTRS Scheduler ... 192
1. Scheduler Graphical Interface ... 192
1.1. Scheduler Not Running Notification ... 192
1.2. Start Scheduler ... 193
2. Scheduler Command Line Interface ... 193
2.1. Unix / Linux ... 194
2.2. Windows ... 195
22. Dynamic Fields ... 198
1. Introduction ... 198
2. Configuration ... 198
2.1. Adding a Dynamic Field ... 199
2.2. Text Dynamic Field Configuration ... 201
2.3. Textarea Dynamic Field Configuration ... 202
2.4. Checkbox Dynamic Field Configuration ... 203
2.5. Dropdown Dynamic Field Configuration ... 204
2.6. Multiselect Dynamic Field Configuration ... 205
2.7. Date Dynamic Field Configuration ... 206
2.8. Date / Time Dynamic Field Configuration ... 207
2.9. Editing a Dynamic Field ... 209
2.10. Showing a Dynamic Field on a Screen ... 210
2.11. Setting a Default Value by a Ticket Event Module ... 217
2.12. Set a Default Value by User Preferences ... 218
2.13. Updating from OTRS 3.0 ... 219
23. Zusätzliche Applikationen ... 222
1. FAQ ... 222
24. Leistungsverbesserung ... 223
1. OTRS ... 223
1.1. TicketIndexModule ... 223
1.2. TicketStorageModule ... 223
1.3. Tickets archivieren ... 224
2. Datenbank ... 225
2.1. MySQL ... 225
2.2. PostgreSQL ... 225
3. Webserver ... 225
3.1. Persistente Datenbankverbindungen ... 225
3.2. Vorgeladene Module - startup.pl ... 225
3.3. Perl Module bei Änderung neu laden ... 226
3.4. Die richtige Strategie wählen ... 226
3.5. mod_gzip/mod_deflate ... 226
25. Datensicherung ... 227
1. Datensicherung ... 227
2. Wiederherstellung der Daten ... 227

A. Weitere Quellen ... 229
1. Homepage OTRS.org ... 229
2. Mailinglisten ... 229
3. Fehler berichten ... 230
4. Kommerzieller Support ... 231
B. Referenz der Konfigurationsoptionen ... 232
1. Framework ... 232
1.1. Core ... 232
1.2. Core::LinkObject ... 241
1.3. Core::Log ... 243
1.4. Core::MIME-Viewer ... 244
1.5. Core::MirrorDB ... 246
1.6. Core::PDF ... 247
1.7. Core::Package ... 250
1.8. Core::PerformanceLog ... 252
1.9. Core::ReferenceData ... 253
1.10. Core::SOAP ... 253
1.11. Core::Sendmail ... 254
1.12. Core::Session ... 257
1.13. Core::SpellChecker ... 261
1.14. Core::Stats ... 263
1.15. Core::Stats::Graph ... 264
1.16. Core::Time ... 268
1.17. Core::Time::Calendar1 ... 273
1.18. Core::Time::Calendar2 ... 275
1.19. Core::Time::Calendar3 ... 278
1.20. Core::Time::Calendar4 ... 281
1.21. Core::Time::Calendar5 ... 284
1.22. Core::Time::Calendar6 ... 287
1.23. Core::Time::Calendar7 ... 290
1.24. Core::Time::Calendar8 ... 292
1.25. Core::Time::Calendar9 ... 295
1.26. Core::Web ... 298
1.27. Core::WebUserAgent ... 311
1.28. Crypt::PGP ... 311
1.29. Crypt::SMIME ... 313
1.30. Frontend::Admin::AdminCustomerUser ... 314
1.31. Frontend::Admin::ModuleRegistration ... 315
1.32. Frontend::Agent ... 329
1.33. Frontend::Agent::Dashboard ... 335
1.34. Frontend::Agent::LinkObject ... 338
1.35. Frontend::Agent::ModuleMetaHead ... 338
1.36. Frontend::Agent::ModuleNotify ... 339
1.37. Frontend::Agent::ModuleRegistration ... 340
1.38. Frontend::Agent::NavBarModule ... 346
1.39. Frontend::Agent::Preferences ... 346
1.40. Frontend::Agent::SearchRouter ... 351
1.41. Frontend::Agent::Stats ... 351
1.42. Frontend::Customer ... 355
1.43. Frontend::Customer::Auth ... 362
1.44. Frontend::Customer::ModuleMetaHead ... 370
1.45. Frontend::Customer::ModuleNotify ... 370
1.46. Frontend::Customer::ModuleRegistration ... 371
1.47. Frontend::Customer::Preferences ... 372

1.48. Frontend::Public ... 375
1.49. Frontend::Public::ModuleRegistration ... 376
2. Ticket ... 376
2.1. Core ... 376
2.2. Core::FulltextSearch ... 377
2.3. Core::LinkObject ... 378
2.4. Core::PostMaster ... 378
2.5. Core::Stats ... 388
2.6. Core::Ticket ... 389
2.7. Core::TicketACL ... 406
2.8. Core::TicketBulkAction ... 407
2.9. Core::TicketDynamicFieldDefault ... 407
2.10. Core::TicketWatcher ... 414
2.11. Frontend::Admin::ModuleRegistration ... 414
2.12. Frontend::Agent ... 423
2.13. Frontend::Agent::CustomerSearch ... 430
2.14. Frontend::Agent::Dashboard ... 432
2.15. Frontend::Agent::ModuleMetaHead ... 435
2.16. Frontend::Agent::ModuleNotify ... 435
2.17. Frontend::Agent::ModuleRegistration ... 436
2.18. Frontend::Agent::Preferences ... 449
2.19. Frontend::Agent::SearchRouter ... 455
2.20. Frontend::Agent::Ticket::ArticleAttachmentModule ... 455
2.21. Frontend::Agent::Ticket::ArticleComposeModule ... 456
2.22. Frontend::Agent::Ticket::ArticleViewModule ... 456
2.23. Frontend::Agent::Ticket::ArticleViewModulePre ... 457
2.24. Frontend::Agent::Ticket::MenuModule ... 458
2.25. Frontend::Agent::Ticket::MenuModulePre ... 464
2.26. Frontend::Agent::Ticket::ViewBounce ... 468
2.27. Frontend::Agent::Ticket::ViewBulk ... 469
2.28. Frontend::Agent::Ticket::ViewClose ... 472
2.29. Frontend::Agent::Ticket::ViewCompose ... 479
2.30. Frontend::Agent::Ticket::ViewCustomer ... 482
2.31. Frontend::Agent::Ticket::ViewEmailNew ... 483
2.32. Frontend::Agent::Ticket::ViewEscalation ... 485
2.33. Frontend::Agent::Ticket::ViewForward ... 487
2.34. Frontend::Agent::Ticket::ViewFreeText ... 489
2.35. Frontend::Agent::Ticket::ViewHistory ... 495
2.36. Frontend::Agent::Ticket::ViewMailbox ... 496
2.37. Frontend::Agent::Ticket::ViewMerge ... 497
2.38. Frontend::Agent::Ticket::ViewMove ... 499
2.39. Frontend::Agent::Ticket::ViewNote ... 501
2.40. Frontend::Agent::Ticket::ViewOwner ... 507
2.41. Frontend::Agent::Ticket::ViewPending ... 514
2.42. Frontend::Agent::Ticket::ViewPhoneInbound ... 520
2.43. Frontend::Agent::Ticket::ViewPhoneNew ... 523
2.44. Frontend::Agent::Ticket::ViewPhoneOutbound ... 527
2.45. Frontend::Agent::Ticket::ViewPrint ... 530
2.46. Frontend::Agent::Ticket::ViewPriority ... 530
2.47. Frontend::Agent::Ticket::ViewQueue ... 537
2.48. Frontend::Agent::Ticket::ViewResponsible ... 540
2.49. Frontend::Agent::Ticket::ViewSearch ... 546
2.50. Frontend::Agent::Ticket::ViewStatus ... 556
2.51. Frontend::Agent::Ticket::ViewZoom ... 557

2.52. Frontend::Agent::TicketOverview 560
2.53. Frontend::Agent::ToolBarModule 563
2.54. Frontend::Customer 567
2.55. Frontend::Customer::ModuleMetaHead 571
2.56. Frontend::Customer::ModuleRegistration 571
2.57. Frontend::Customer::Preferences 574
2.58. Frontend::Customer::Ticket::ViewNew 575
2.59. Frontend::Customer::Ticket::ViewPrint 580
2.60. Frontend::Customer::Ticket::ViewSearch 580
2.61. Frontend::Customer::Ticket::ViewZoom 583
2.62. Frontend::Queue::Preferences 586
2.63. Frontend::SLA::Preferences 587
2.64. Frontend::Service::Preferences 587
C. Danksagungen 588
D. GNU Free Documentation License 590
0. PREAMBLE 590
1. APPLICABILITY AND DEFINITIONS 590
2. VERBATIM COPYING 591
3. COPYING IN QUANTITY 591
4. MODIFICATIONS 592
5. COMBINING DOCUMENTS 593
6. COLLECTIONS OF DOCUMENTS 593
7. AGGREGATION WITH INDEPENDENT WORKS 594
8. Übersetzung 594
9. TERMINATION 594
10. FUTURE REVISIONS OF THIS LICENSE 594
How to use this License for your documents 594

Tabellenverzeichnis

3.1. Die folgenden Perl-Module werden von OTRS benötigt. 18
3.2. Die verschiedenen Skripte für die cron-Jobs von OTRS. 28
5.1. Standardmäßig vorhandene Gruppen in OTRS 48
5.2. Mit OTRS-Gruppen verknüpfte Rechte 50
5.3. Ereignisse für automatische Antworten 69
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 229

Beispiele

5.1. Aussortierung von Spammails in eine bestimmte Queue 79
7.1. .fetchmailrc 91
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 92
11.1. Konfiguration eines Datenbank-Kunden-Backends 99
11.2. Firmen-Tickets mit einem Datenbank-Backend 102
11.3. Konfiguration eines LDAP Kunden-Backends 102
11.4. Verwendung von Firmentickets mit einem LDAP-Backend 104
11.5. Gleichzeitige Einbindung mehrerer verschiedener Kunden Backends 104
11.6. Agentenauthentifizierung gegen ein Datenbank-Backend 107
11.7. Agentenauthentifizierung gegen ein LDAP Backend 107
11.8. Agentenauthentifizierung über HTTPBasic 108
11.9. Agentenauthentifizierung gegen ein Radius-Backend 109
11.10. Kundenauthentifizierung gegen ein Datenbank-Backend 109
11.11. Kundenauthentifizierung gegen ein LDAP Backend 109
11.12. Kundenauthentifizierung über HTTPBasic 110
11.13. Kundenauthentifizierung gegen ein Radius-Backend 110
18.1. ACL, die das Verschieben von Tickets in eine Queue nur bei einer Ticket-Priorität von 5 erlaubt. 128
18.2. ACL allowing movement into a queue of only those tickets with ticket priority 5 stored in the database. 129
18.3. ACL, die das Schließen von Tickets in der raw-Queue verbietet und den Schließen-Schalter ausblendet. 129
18.4. ACL removing always state closed successful. 130
18.5. ACL only showing Hardware services for tickets that are created in queues that start with "HW". 130
18.6. Reference showing all possible important ACL settings. 131
19.1. Festlegen einer Wertereihe bei einem Element 143
19.2. Festlegen einer Wertereihe bei zwei Elementen 143
21.1. Example To Start The OTRS Scheduler Form An Init.d Script 194
21.2. Example To Start The OTRS Scheduler 195
21.3. Example To Force Stop The OTRS Scheduler 195
21.4. Example To Register The OTRS Scheduler Into the Widows SCM 196
21.5. Example To Start The OTRS Scheduler 196
21.6. Example To Force Stop The OTRS Scheduler 197
22.1. Activate Field1 in New Phone Ticket Screen. 211
22.2. Activate Field1 in New Phone Ticket Screen as mandatory. 212
22.3. Activate several fields in New Phone Ticket Screen. 213
22.4. Deactivate some fields in New Phone Ticket Screen. 214
22.5. Activate Field1 in Ticket Zoom Screen. 215
22.6. Activate Field1 in Ticket Overview Small Screens. 216
22.7. Activate Field1 in TicketCreate event. 217
22.8. Activate Field1 in the User preferences. 218

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. Bitte teilen 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 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.

1. 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 eingesparen 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 einzeln 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.

2. 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 eineMehrfachdosierung 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 Kundens 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 ganz normale 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.

1. 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.

2. 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 verschidenen 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.

2.1. New features of OTRS 3.1

2.1.1. 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.

- ~~~~~~~~~~~~ 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.

2.1.1.1. 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.

2.1.1.2. 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).

2.1.2. 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.

2.1.3. TICKET MANAGEMENT IMPROVEMENTS

2.1.3.1. Ticket creation improved

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

2.1.3.2. Inbound phone call support

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

2.1.3.3. 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.

2.1.3.4. 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).

2.1.3.5. 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.

2.1.3.6. Configurable Reject Sender Email Address

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

2.1.4. PROCESS AUTOMATION

2.1.4.1. Escalation events added

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

2.1.4.2. 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.

2.1.4.3. 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.

2.1.4.4. GenericAgent improved

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

2.1.5. USER INTERFACE, RICH TEXT EDITOR, CHARSET

2.1.5.1. User interface performance improved

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

2.1.5.2. 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 and CKEditor 3.6.
- IE9 support improved.
- Resizable dialogs.

2.1.5.3. 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.

2.1.6. DATABASE DRIVER SUPPORT

2.1.6.1. PostgreSQL DRIVER compatibility improved

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

2.1.6.2. 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).

2.1.7. MAIL INTEGRATION

2.1.7.1. 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.

2.2. 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 Anwendungszenario 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.

2.3. 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-Tickt 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).

3. 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

3.1. Perl support

- Perl 5.8.8 oder höher

3.2. 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

3.3. 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.

3.4. 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.

4. 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: [illegible].

5. 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: [illegible].

OTRS Group provides subscription support services, customization, consulting and training for [illegible] and [illegible]. It also provides [illegible]. With these Editions, OTRS Group helps organizations to design, deploy and optimize OTRS for each unique environment. Additionally, OTRS Group provides hosted versions including [illegible] and [illegible].

You can find more detailed information about OTRS Group on [illegible] and you can contact us via email on [illegible].

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.

1. 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 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.

1.1. 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 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 'rcapache2 start' 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:~ #
```

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:~ #
```

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:~ # rcapache2 restart
Shutting down httpd                                        done
Starting httpd [ PERL ]                                    done
linux:~ #
```

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

1.2. 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 .

1.3. 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 .

1.4. 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) .

1.5. Installation von OTRS auf Microsoft Windows-Systemen

Die Installation von OTRS unter Microsoft Windows ist denkbar einfach. Laden Sie den auf 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 undcron 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.

2. Manuelle Installation (Linux, Unix)

2.1. 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 .

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#
```

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#
```

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
```

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 alsootrs.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 Komando otrs.SetPermissions.pl /opt/otrs --otrs-user=otrs --web-user=wwwrun --otrs-group=nogroup --web-group=www ein, um die korrekten Zugriffsrechte zu setzen.

2.2. 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.

Name	Beschreibung
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#
```

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.plperl -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 Webservers 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#
```

2.3. Konfiguration des Apache-Webservers

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

Webservers 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 Webservers 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#
```

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*
```

Wenn Sie das o. a. geeignete Startskipt 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.

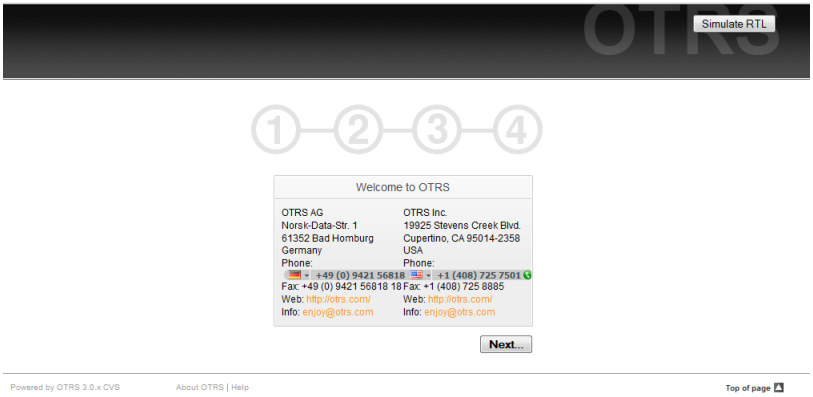
2.4. Einrichten der Datenbank

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

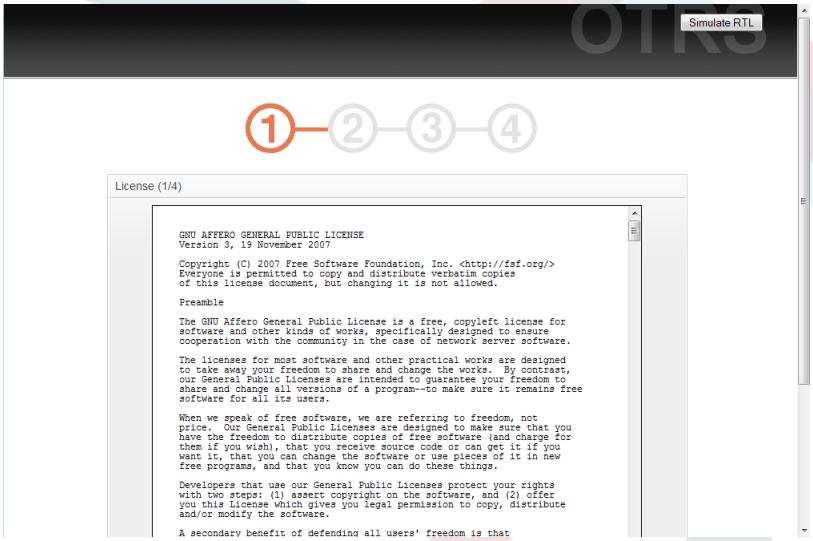
Wenn Sie MySQL als Datenbank-Backend verwenden, können Sie den Web-Installer von OTRS verwenden: .

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

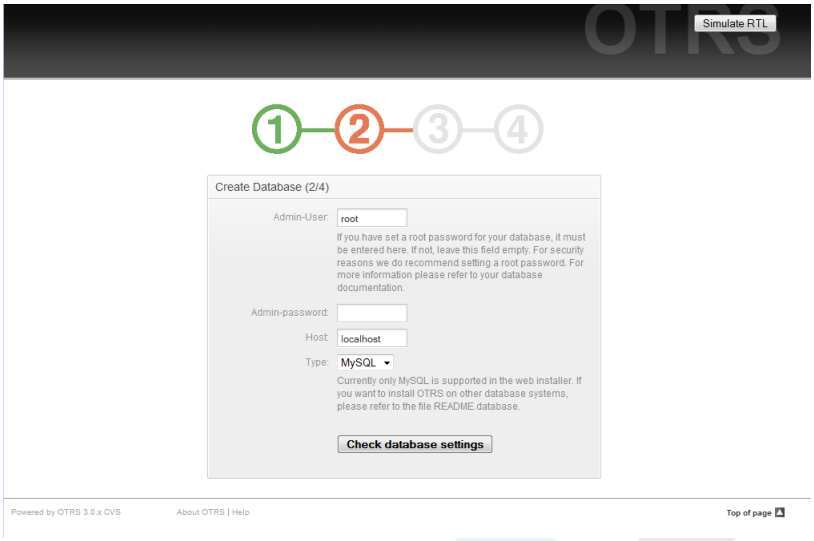
1. Lesen Sie die Informationen über die OTRS-Niederlassungen und klicken Sie auf "Weiter".



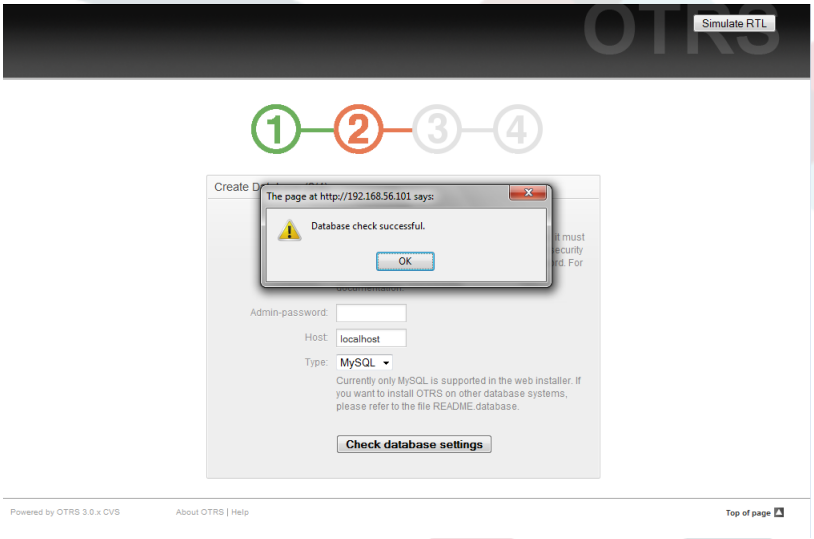
2. Lesen Sie die GNU Affero General Public License und akzeptieren Sie sie durch Klick auf "Akzeptieren".



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.



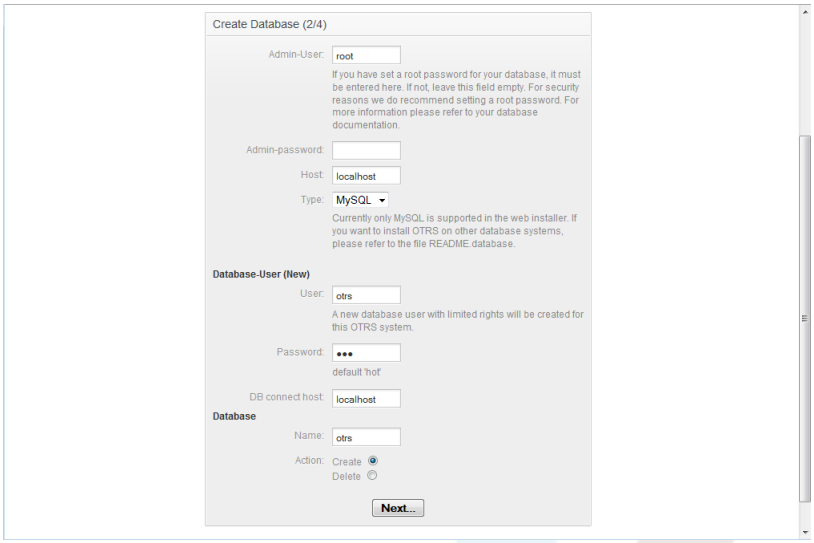
Wenn die Prüfung erfolgreich war, erhalten Sie einen Hinweis. Drücken Sie "Ok" zum Fortfahren.



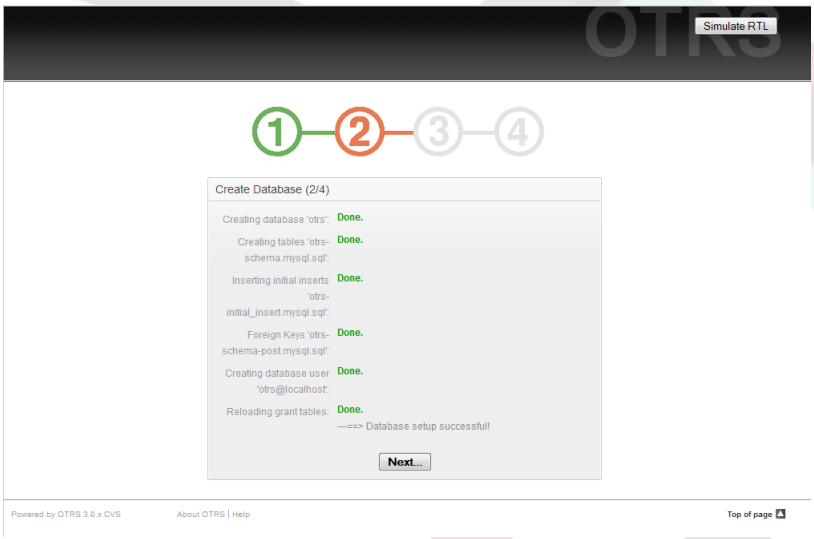
4. Erstellen Sie einen neuen Datenbankbenutzer, wählen Sie einen Namen für die Datenbank und klicken Sie auf "Weiter".

Warnung

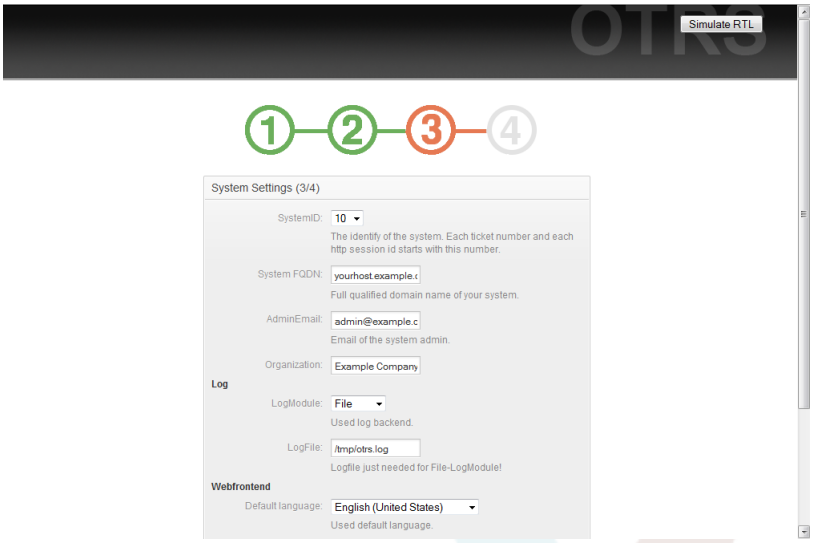
Man sollte nie Standard-Passwörter verwenden. Bitte ändern Sie das Standardpasswort der OTRS-Datenbank!



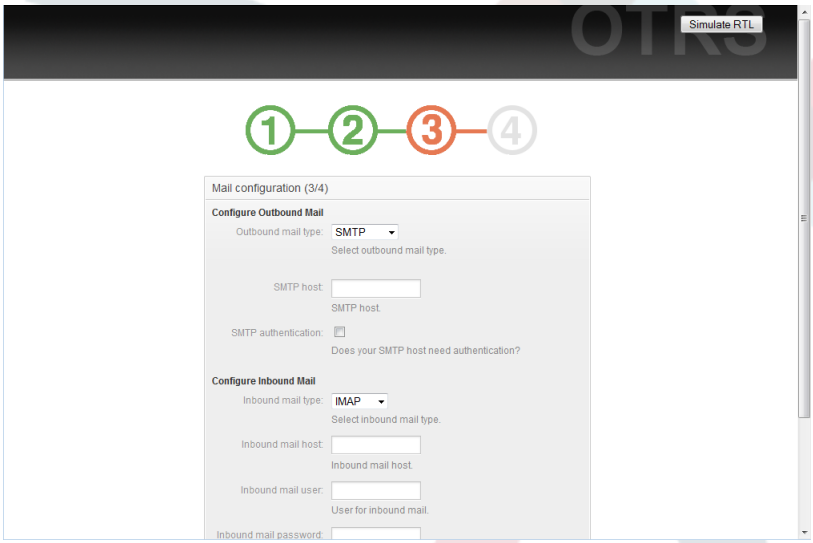
Wenn der Datenbankbenutzer und die Datenbank erfolgreich angelegt wurden, erhalten Sie eine Bestätigung. Klicken Sie auf 'Weiter', um fortzufahren.



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



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



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)
```

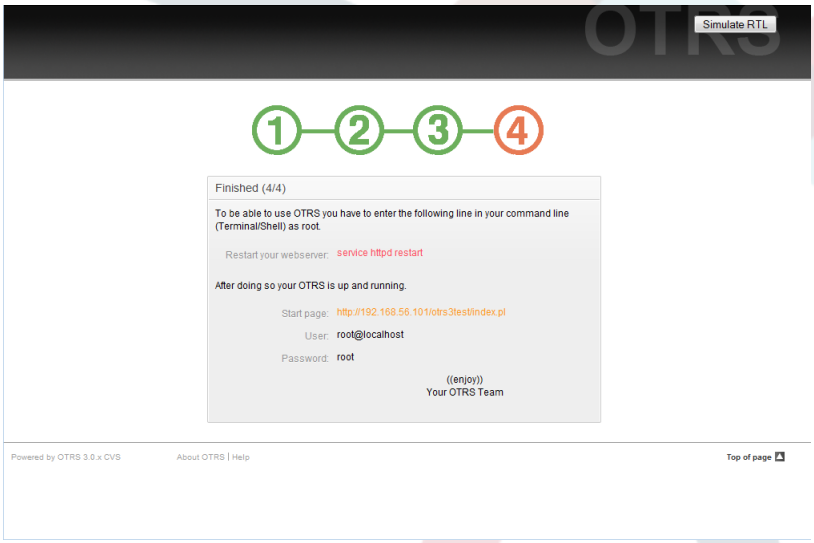
```
 Starting Apache ... done.
 Starting MySQL ... done.
Starting OTRS
 Checking Apache ... done.
 Checking MySQL ... done.
 Checking database connect... (It looks Ok!).
 Enable /opt/otrs/bin/otrs.PostMaster.pl ... done.
 Checking otrs spool dir...  done.
 Creating cronjobs (source /opt/otrs/var/cron/*)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

  -->> http://linux.example.com/otrs/index.pl <<--
                                                                done
                                                                done
linux:~ #
```

Herzlichen Glückwunsch! Die Installation ist abgeschlossen, jetzt können Sie mit dem System arbeiten. Um sich anzumelden, verwenden Sie bitte die Adresse in ihrem Webbrowser. Melden Sie sich als Administrator and, 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.



2.4.2. 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#
```

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';
```

2.5. 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 Homeverzeichnisses 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#
```

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#
```

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#
```

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#
```

3. Upgrade des OTRS Frameworks

These instructions are for people upgrading OTRS from version ⁓⁓⁓ to ⁓⁓⁓, 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.

Within a single minor version you can skip patch level releases if you want to upgrade. For instance you can upgrade directly from OTRS 3.1.1 to version 3.1.4. If you need to do such a "patch level upgrade", you should skip steps 9, 11 and 13-19.

Please note that for upgrades from 3.1.beta1 or 3.1.beta2, an additional step 21 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/*`
   - as well as the database

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

4. Setup new system (optional)

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

5. 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.

6. 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).

7. 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.

8. 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
```

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

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

10. 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.

11. 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.

12. Installierte Pakete überprüfen

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

13 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.

14 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.

15 TicketHistory

The TicketHistory type TicketFreeTextUpdate was renamed to TicketDynamicFieldUpdate. If you have any custom reporting using this, please adjust it.

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 ACL changes

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.

Please also note that since OTRS 3.1.5, it is possible to match either the values of an existing ticket as it is on the database (new setting "PropertiesDatabase"), in addition to the existing match against the current values of the form. Details can be found at http://doc.otrs.org/3.1/en/html/acl.html.

20 SMIME file names

This step is ONLY needed if you use SMIME to encrypt/sign your emails. You need to take this step if you upgrade from 3.0.x OR if you upgrade from 3.1.x before 3.1.6 while using OpenSSL 1.x to encrypt/sign your emails.

Between OTRS 3.0.x and 3.1.x, the file naming of private secret file names changed and need to be updated. In addition, OpenSSL 0.9.x and 1.x use different algorithms to calculate the certificate hash. To update your private secrets and certificate file names, please run this command:

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

This step must also be taken if you change a running installation from OpenSSL 0.9 to 1.x.

21Database 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
```

22Gut gemacht!

4. 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 neuene Version, der Wiederherstellung der Datenbank und dem Ausführen der Upgrade-Prozedur.

Upgrading is described in FAQ# 4200351, and there is also an informative YouTube video available.

5. Upgrading Microsoft SQL Server Data Types

Starting OTRS version [illegible], OTRS uses the [illegible] data type rather than [illegible] or [illegible], to store textual data. This is because the [illegible] 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 [illegible] data type is deprecated since [illegible]. Due to this, starting with OTRS version 3.1, the minimal SQL Server version required for operation with OTRS is now [illegible].

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.

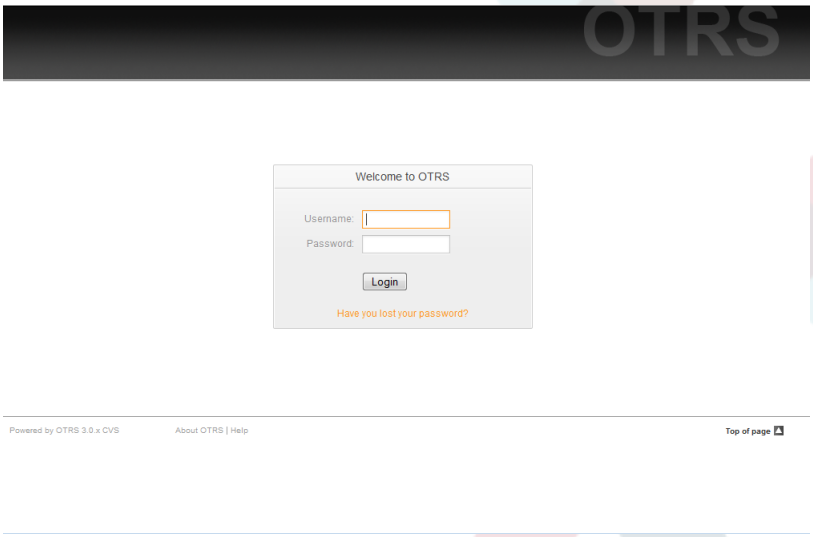
Kapitel 4. Erste Schritte

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.

1. 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.

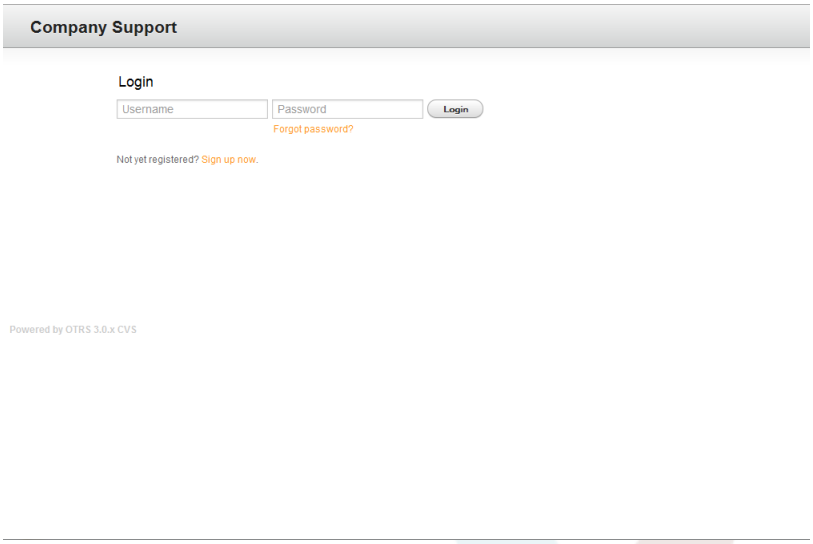
Supposing your OTRS host is reachable via the URL , then the OTRS login screen can be reached by using the address in a web browser (see Figure below).



2. 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.

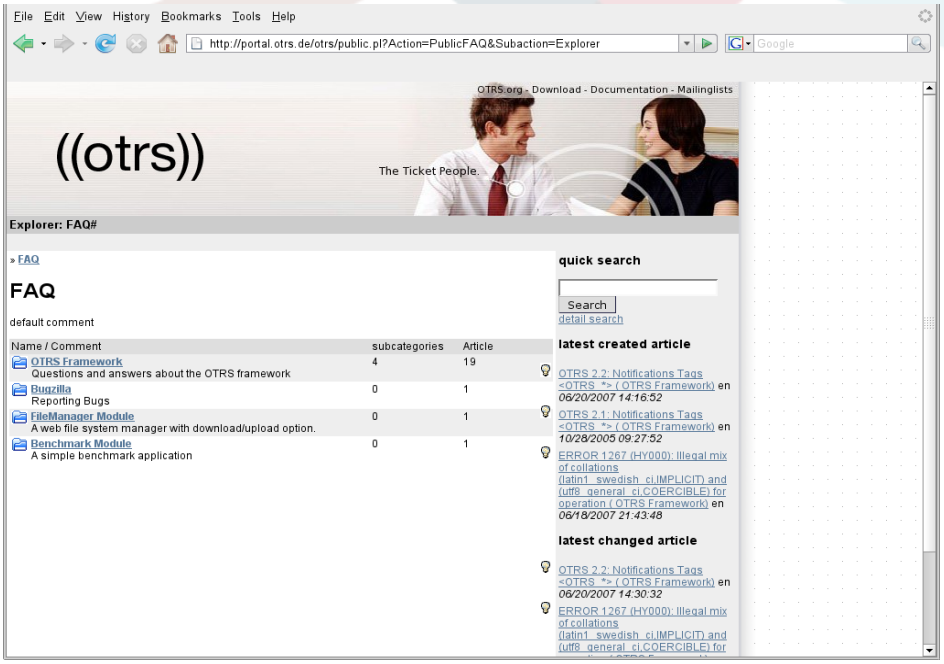
Gemäß dem obigen Beispiel kann der Login-Bildschirm für die Kunden über die URL geladen werden.



3. 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 erreichbar. Ebenfalls ist ein Zugriff auf die FAQ über die URL möglich



4. Die erste Anmeldung

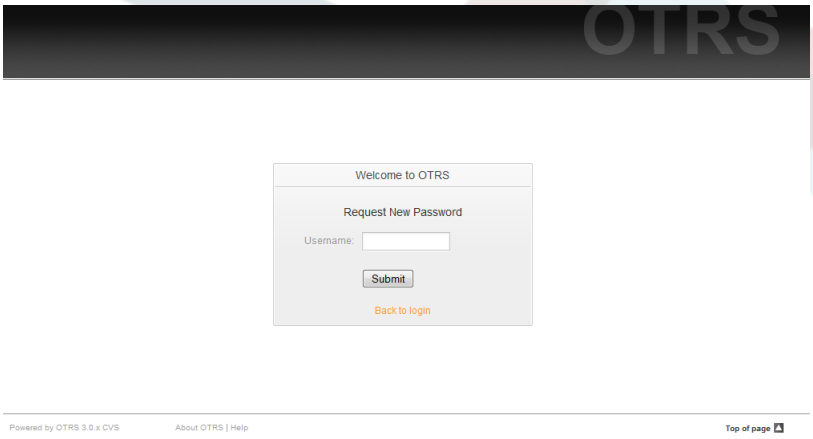
Der Zugriff auf den Login-Bildschirm wird im Abschnitt Agenten-Weboberfläche beschrieben. Hier haben Sie die Möglichkeit, einen Benutzernamen und ein Kennwort anzugeben. Um sich als OTRS-Administrator anzumelden, verwenden Sie als Benutzername "root@localhost" und als Kennwort "root".

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.

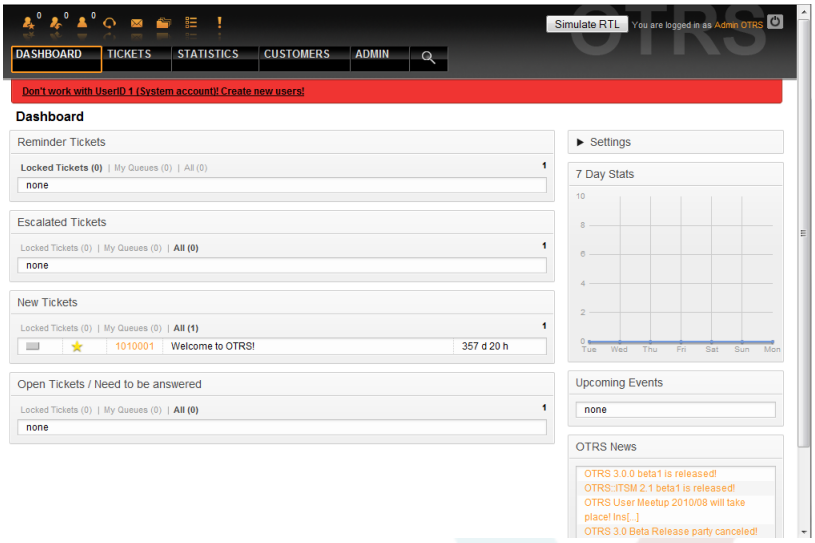
Wollen Sie sich nicht als OTRS-Administrator anmelden, geben Sie einfach den Benutzernamen und das Kennwort Ihres normalen OTRS-Accounts in die dafür vorgesehenen Eingabefelder ein.

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.



5. 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.



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. Duch 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.

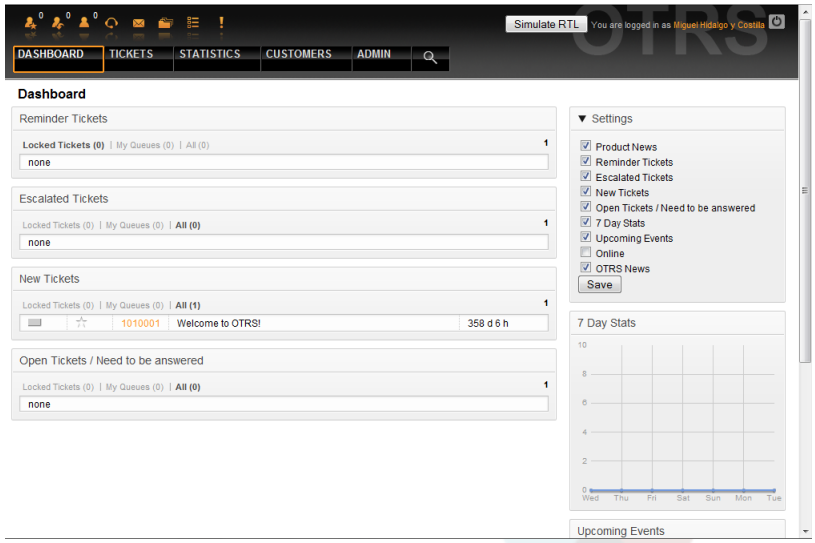
Die rote Leiste unterhalb des Navigationsbereiches zeigt verschiedene Systemnachrichten an. Wenn Sie als OTRS-Administrator-Benutzer angemeldet sind, erhalten Sie hier den Hinweis, dass Sie mit diesem Benutzer nicht regulär arbeiten sollten.

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" ligen. "Meine Queues" sind Queues die Sie in Ihrer Benutzer-Konfiguration markieren können als Queues, die speziell überwacht werden sollen.

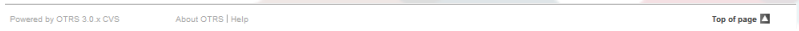
Auf der rechten Seite befindet sich der Bereich "Einstellungen". Klicken Sie darauf um den Bereich zu öffnen und die verschiedenen Einstellmöglichkeiten zu sehen. Sie können dann

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



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.



6. 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.

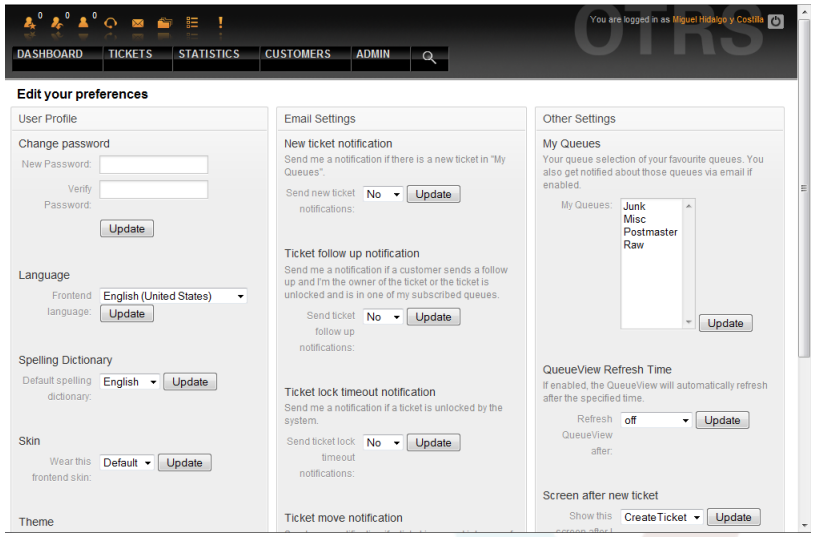
Weil OTRS Zugriffskontrolle für Agenten, Gruppen und Rollen unterstützt, ist es denkbar einfach, Queues einzurichten, die nur von bestimmten Agenten benutzt werden können. Herr Mustermann könnte auch ein anderes Szenario umsetzen, um seine Tickets in die verschiedenen Queues zu bekommen, mit Hilfe von Filterregeln, oder falls zwei verschidenene 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.

7. 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.



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

Benutzerprofil

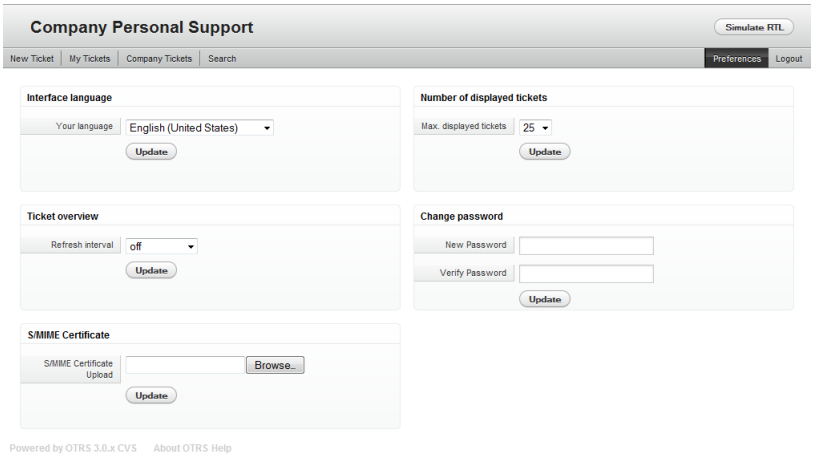
- Aktuelles Passwort ändern.
- Sprache der Benutzeroberfläche wechseln.
- Oberflächen-Skin auswählen.
- Oberflächen-Theme auswählen.
- Abwesenheitszeit aktivieren und konfigurieren.

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.



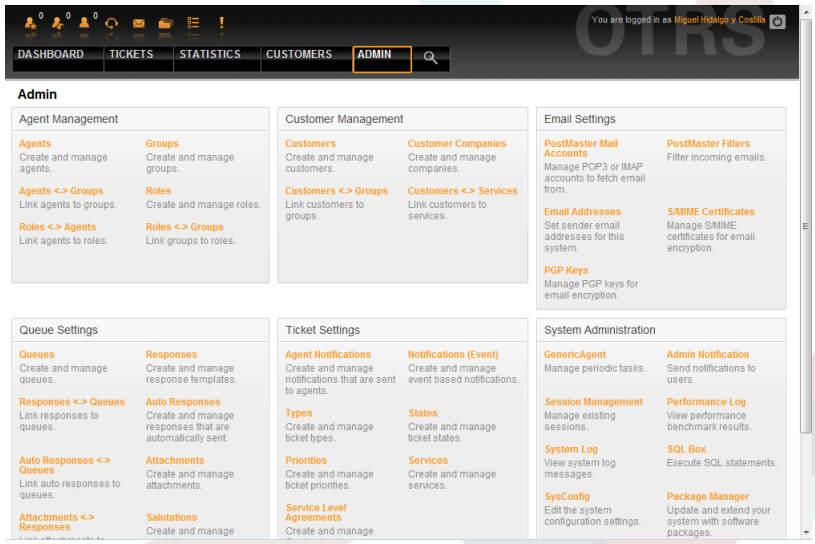
Ein Kunde kann die Sprache der Benutzeroberfläche wechseln, die Aktualisierungszeit der Ticketübersichten verändern und die max. Anzahl angezeigter Tickets pro Seite einstellen. Zusätzlich kann das Passwort geändert werden.

Kapitel 5. Der Administrationsbereich von OTRS

1. Grundlagen

OTRS-Administratioren 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.



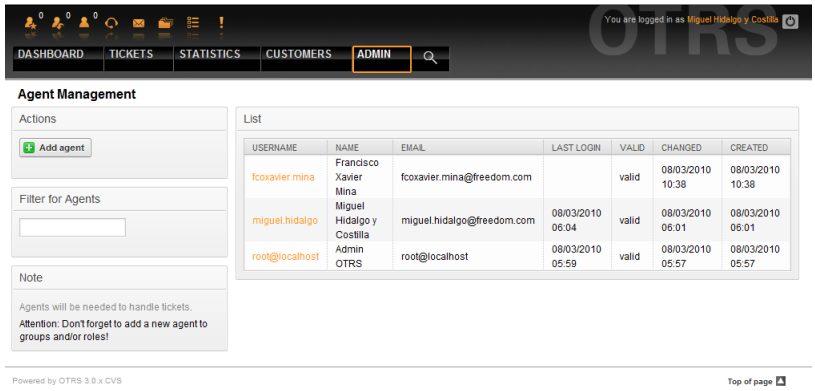
2. Benutzer, Gruppen und Rollen

2.1. Benutzer

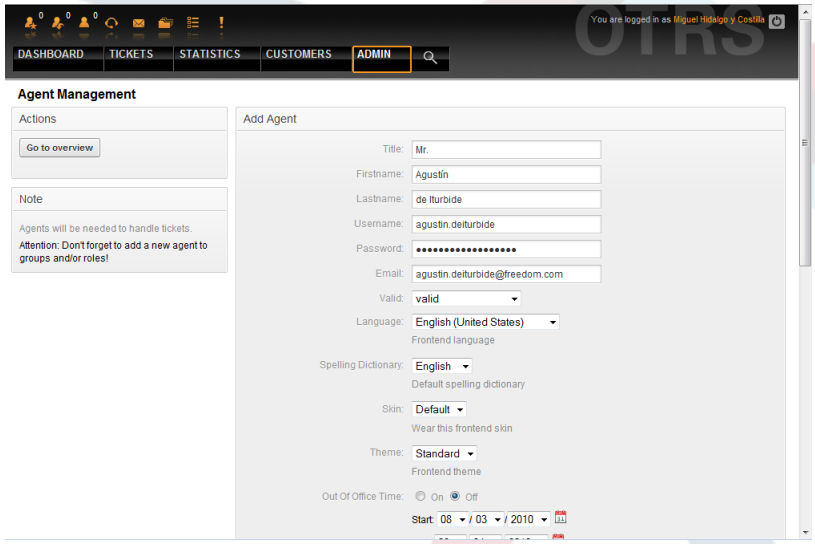
Über den Link "Benutzer" gelangen Sie in die Benutzerverwaltung von OTRS. Hier können Sie Benutzer anlegen, bearbeiten und deaktivieren. Weiterhin lassen sich einige grundlegende Einstellungen für den Benutzer festlegen, z. B. die Oberflächensprache oder das Anzeigeschema.

Anmerkung

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



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



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.

2.2. Gruppen

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

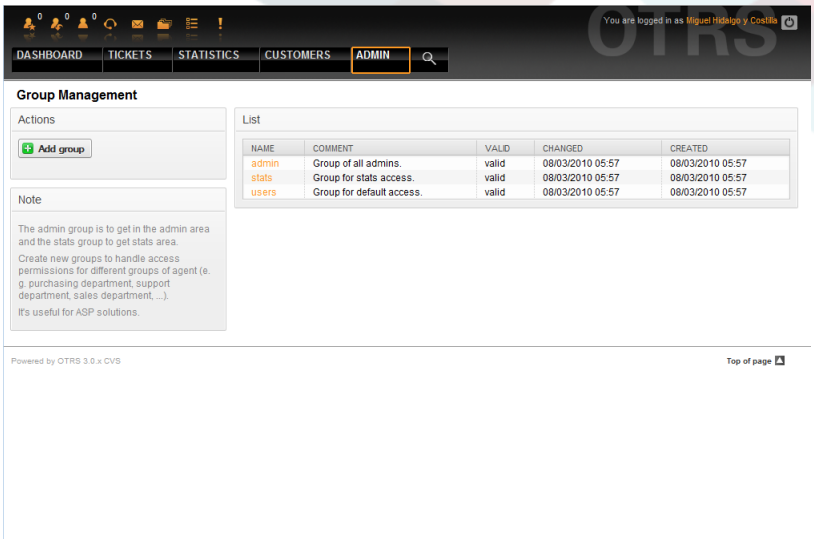
Tabelle 5.1. Standardmäßig vorhandene Gruppen in OTRS

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".

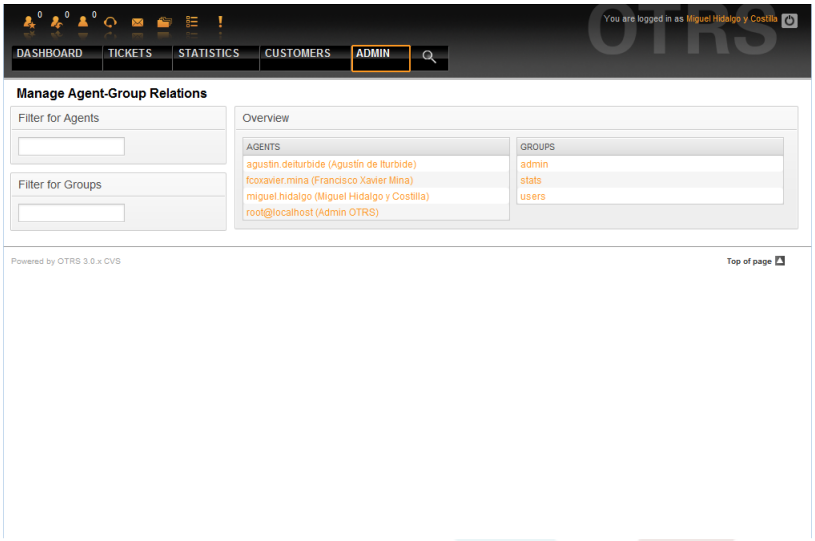
Sie können durch Klick auf "Gruppen" im Administrationsbereich in die Gruppenverwaltung gelangen.



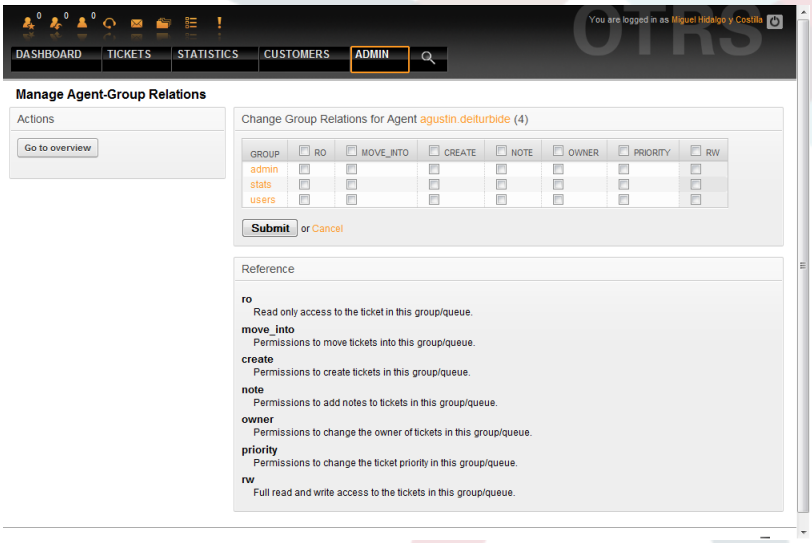
Anmerkung

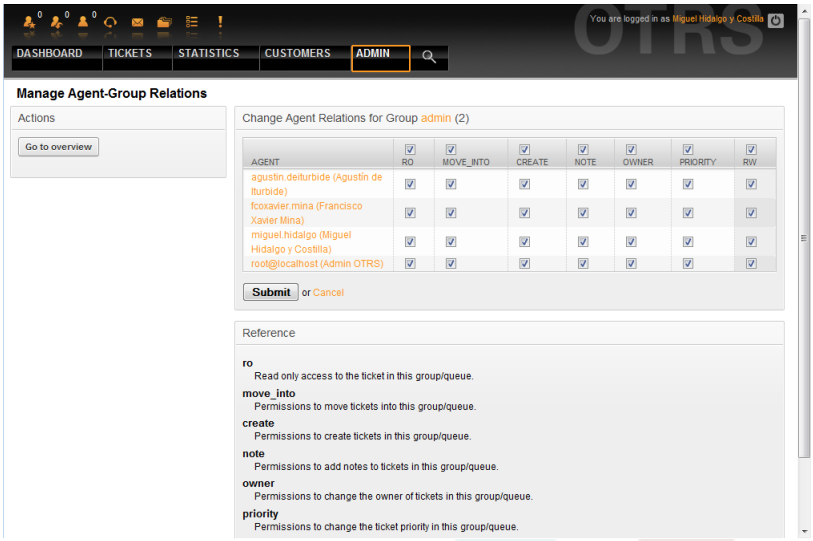
In OTRS können Gruppen deaktiviert, aber nicht gelöscht werden. Deaktivieren Sie eine Gruppe, indem Sie für "Gültig" den Wert entweder auf "ungültig" oder auf "ungültig-temporär" setzen.

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



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.





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 -Rechte hat, d. h. Tickets, an denen der Benutzer

arbeiten muss. Wenn Sie dieses Verhalten ändern wollen, können Sie Ticket::Frontend::AgentTicketQueue###ViewAllPossibleTickets auf ~~ setzen.

2.3. 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.

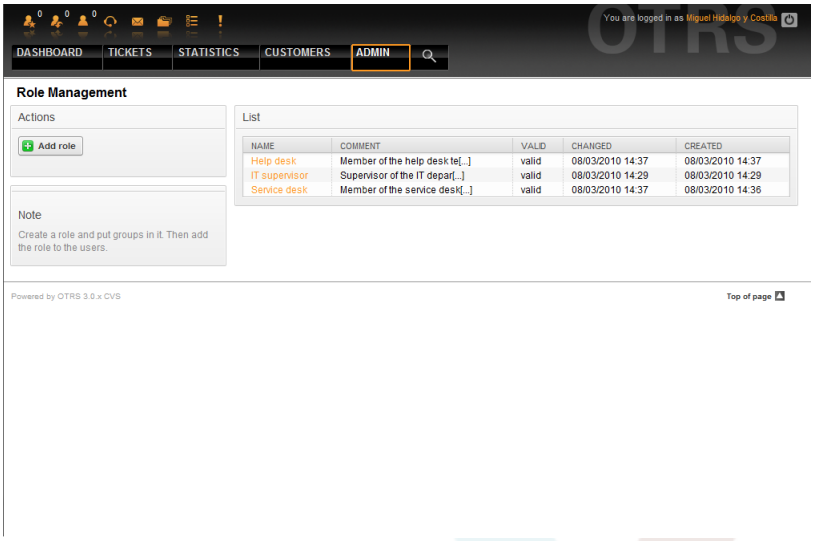
Im Rahmen einer Unternehmensumstrukturierung wird eines Tages zusätzlich eine Abteilung eröffnet, die Produkte verkaufen soll. Es müssen Angebote, Auftragsbestätigungen und Rechnungen erstellt, Anfragen bearbeitet, Bestellungen ans Lager weitergeleitet und Stornierungen entgegen genommen werden. Ein 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!

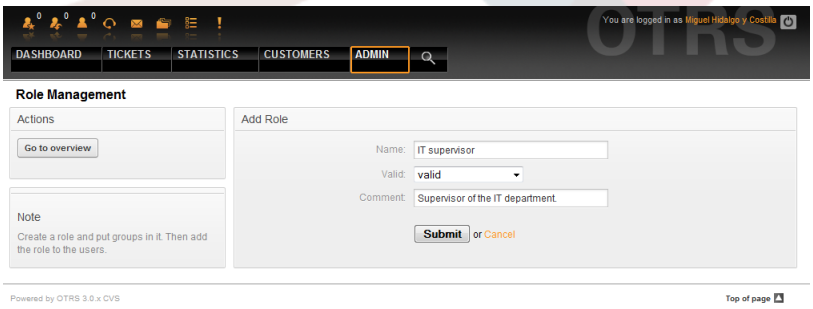
Sie können die Rollenverwaltung durch Klick auf "Rollen" im Administrationsbereich erreichen.



Anmerkung

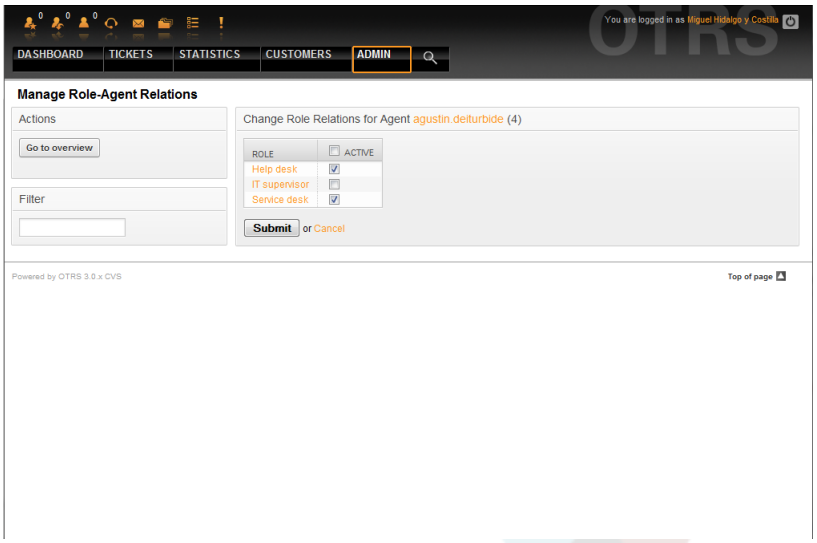
In OTRS können Rollen deaktiviert, aber nicht gelöscht werden. Deaktivieren Sie eine Rolle, indem Sie für "Gültig" den Wert entweder auf "ungültig" bzw. "ungültig-temporär" setzen.

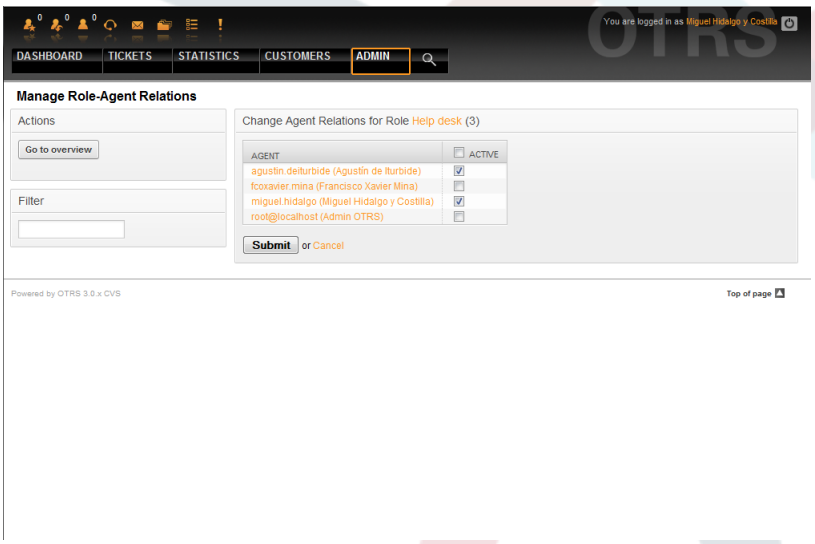
Eine Übersicht aller Rollen im System erscheint, um einen Eintrag zu bearbeiten, klicken Sie einfach auf den Rollennamen. In einer Neuinstallation gibt es keine vordefinierten Rollen. Um eine anzulegen, klicken Sie auf den "Rolle hinzufügen"-Knopf und geben Sie die erforderlichen Daten an.



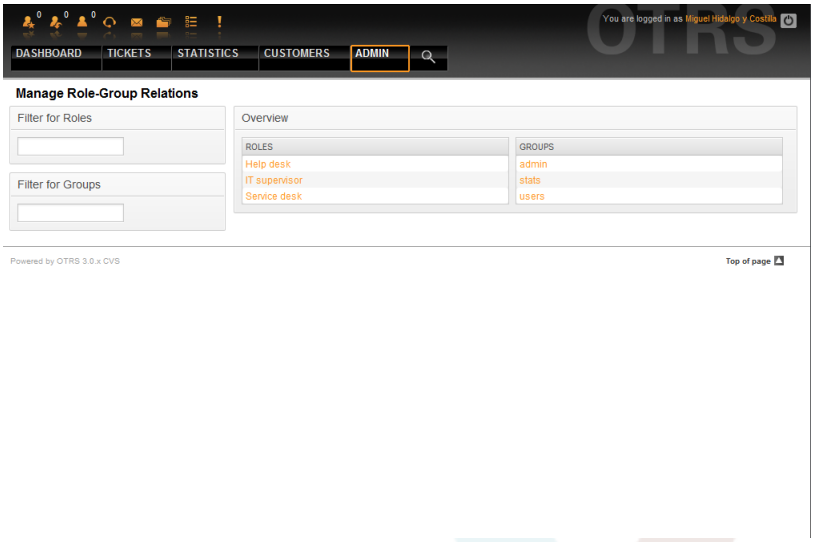
Um einen Überblick über alle Rollen und Agenten im System zu erhalten, klicken Sie auf den Link "Rollen <-> Benutzer" im Administrationsbereich. Sie könnnen die Filter verwenden, um einen bestimmten Eintrag zu finden. Wenn Sie die Rollenzordnungen eines Benutzers

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

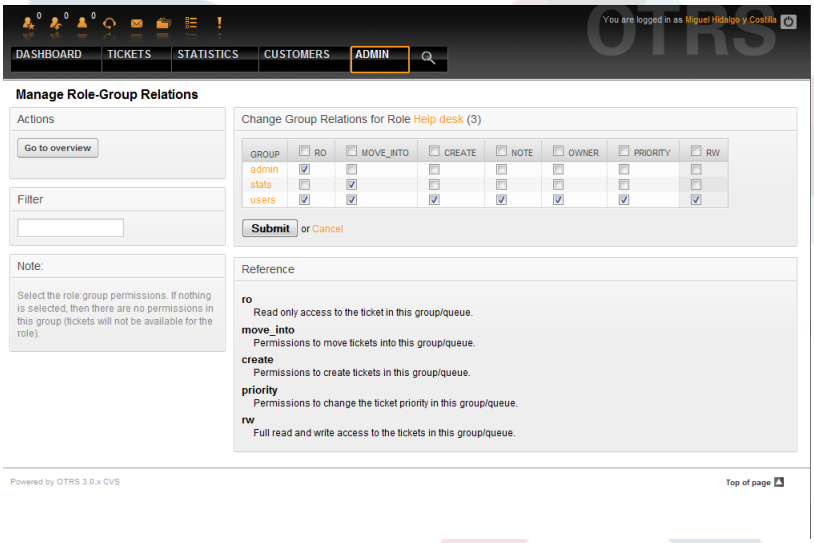


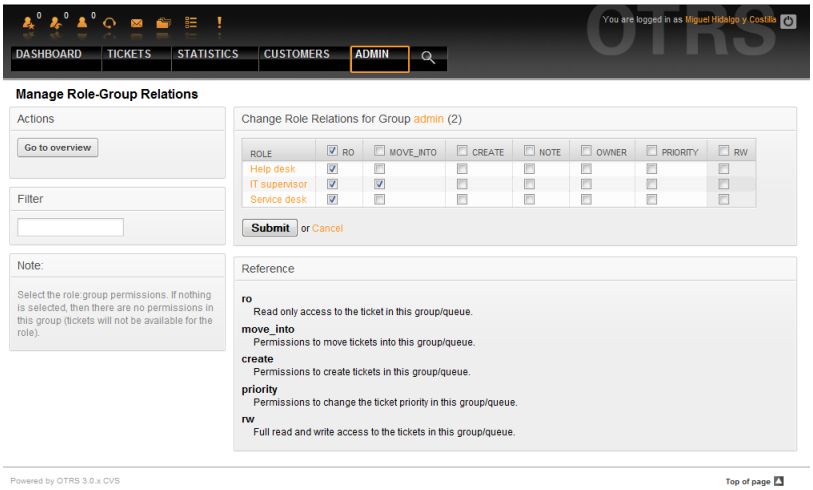


Um eine Übersicht aller Rollen und Gruppen im System zu erhalten, verwenden Sie den Link "Rollen <-> Gruppen" im Administrationsbereich.



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

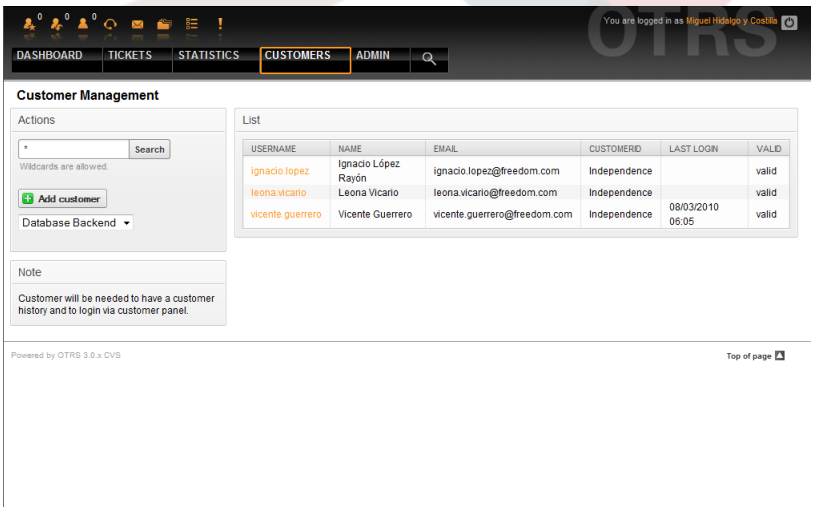




3. Kundenbenutzer und Kundengruppen

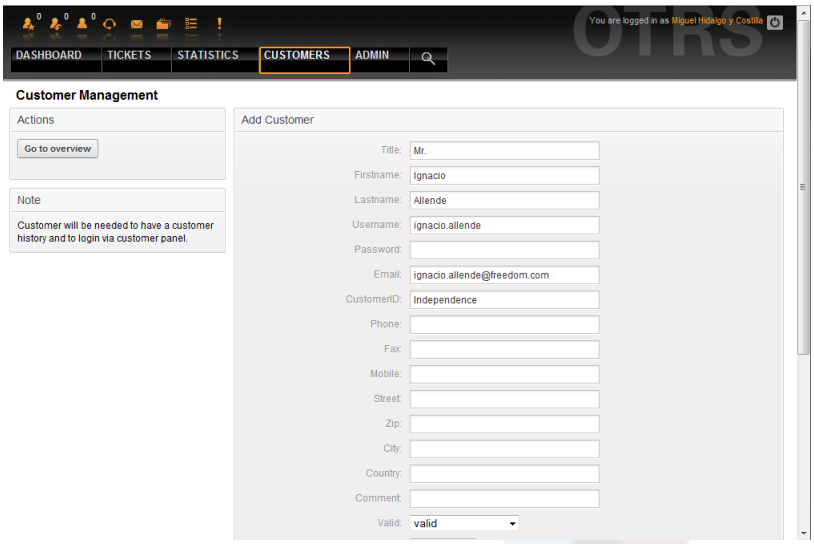
3.1. Kunden

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.



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.



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

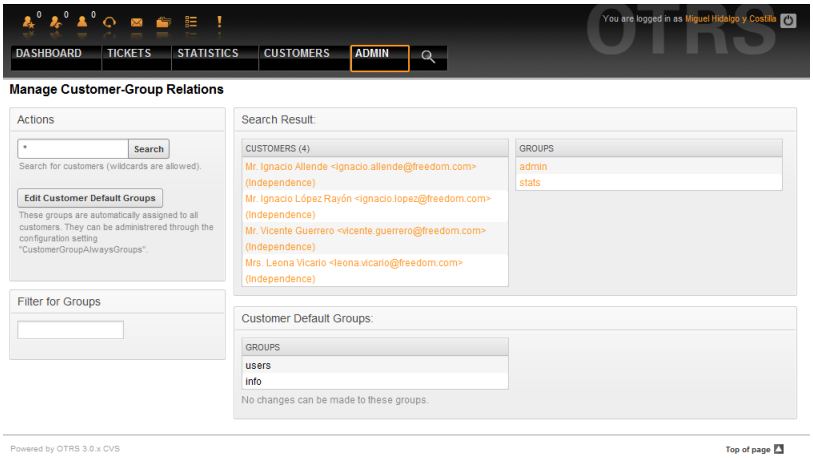
In OTRS können Kunden deaktivert, aber nicht gelöscht werden. Sie können einen Kunden deaktivieren, indem Sie den Wert für "Gültig" auf "ungültig" oder "ungültig-temporär" setzen.

3.2. Kundengruppen

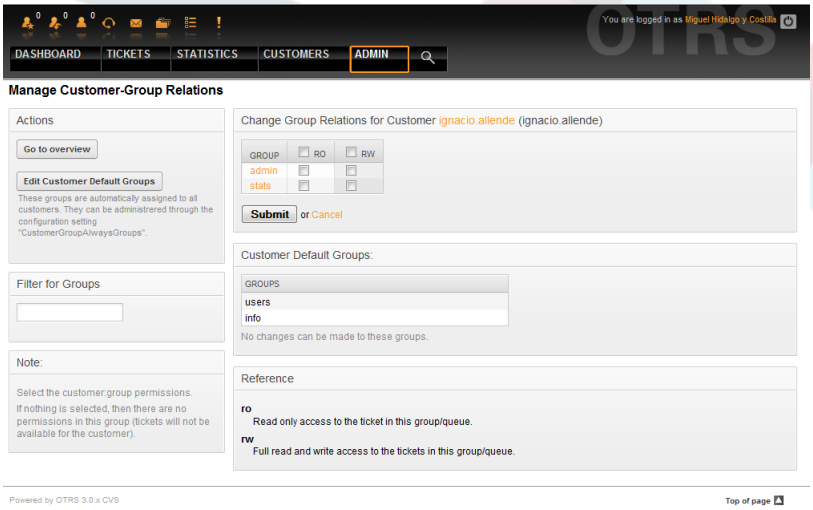
Kunden können auch Gruppen zugeordnet werden. Das ist nützlich, wenn Kunden nur auf bestimmte Queues zugreifen können sollen. Legen Sie zuerst die benötigten Gruppen in der Gruppenverwaltung an . Fügen Sie dann die Queues hinzu und ordnen Sie diese der neuen Gruppe zu.

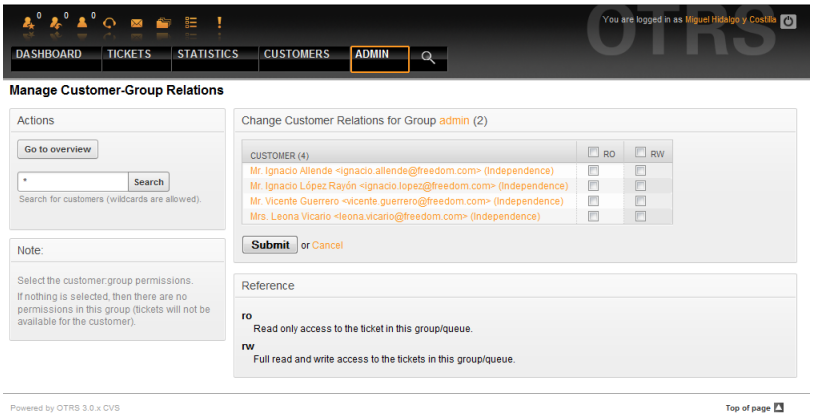
Im nächsten Schritt aktivieren Sie mit Hilfe des Konfigurationsparameters CustomerGroupSupport die Unterstützung für Kundengruppen. Mit Hilfe des Parameters CustomerGroupAlwaysGroups legen Sie fest, welchen Gruppen ein neu angelegter Kundenbenutzer automatisch zugeordnet werden soll.

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



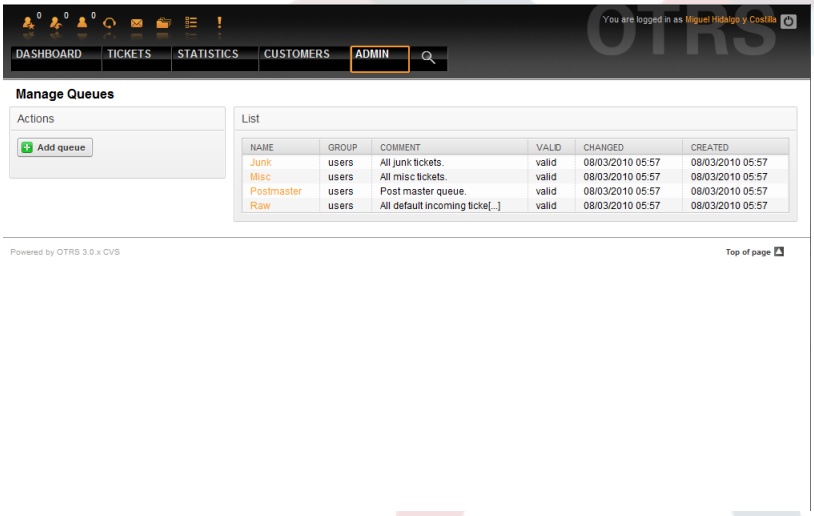
Klicken Sie auf den Kunden- oder Gruppennamen, um die jeweiligen Zuordnungen zu bearbeiten.



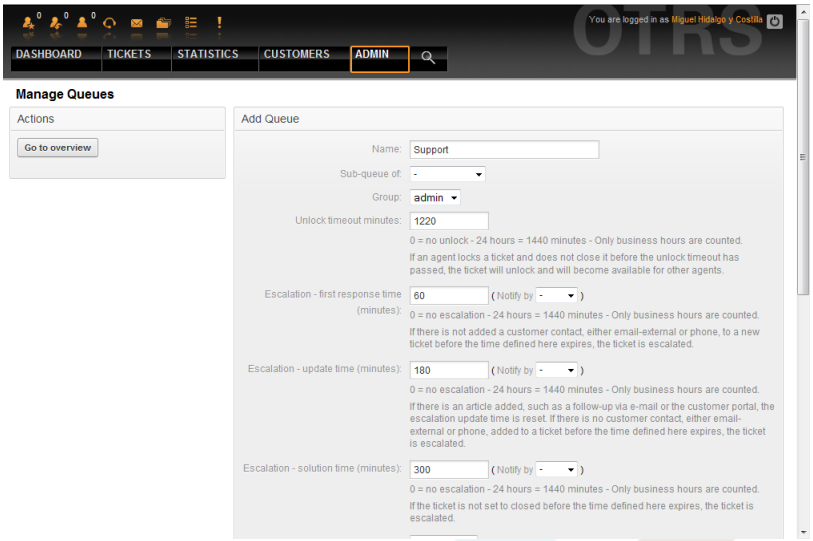


4. Queues

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.



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.



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.

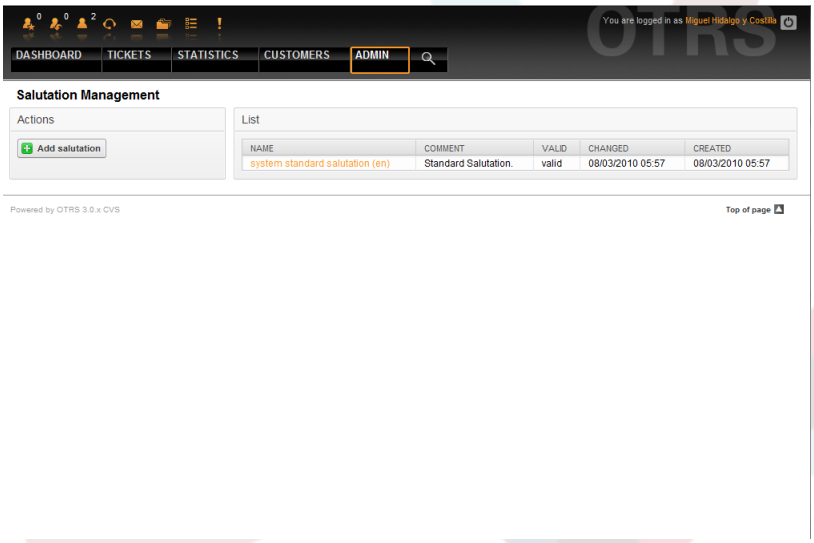
Anmerkung

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.

5. Anreden, Signaturen, Anlagen und Antwortvorlagen

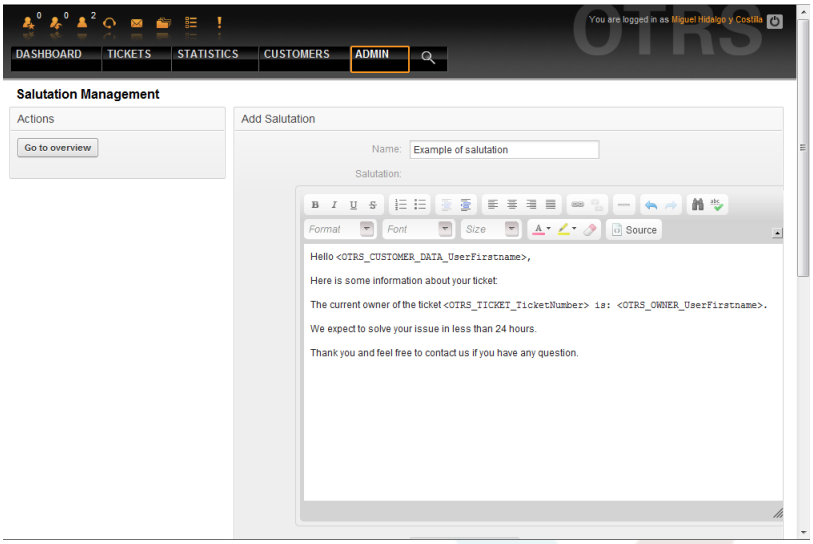
5.1. Anreden

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.



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.



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

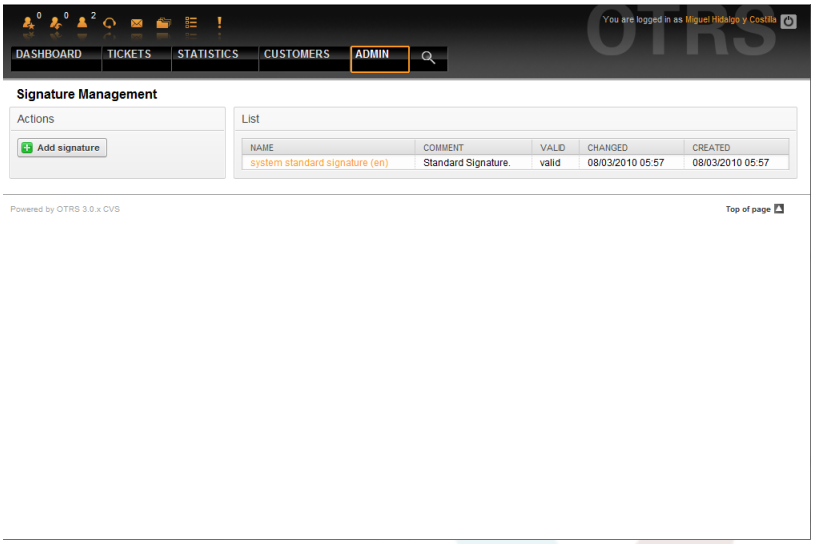
Im unteren Teil der Bildschirmmaske zum Ändern oder Anlegen einer Anrede finden Sie die verschiedenen OTRS-Variablen, die Sie für die dynamische Aufbereitung eines Anredentextes verwenden können. Bauen Sie z. B. in Ihrem Anredentext die Variable <OTRS_LAST_NAME> ein, so wird diese später durch den Nachnamen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

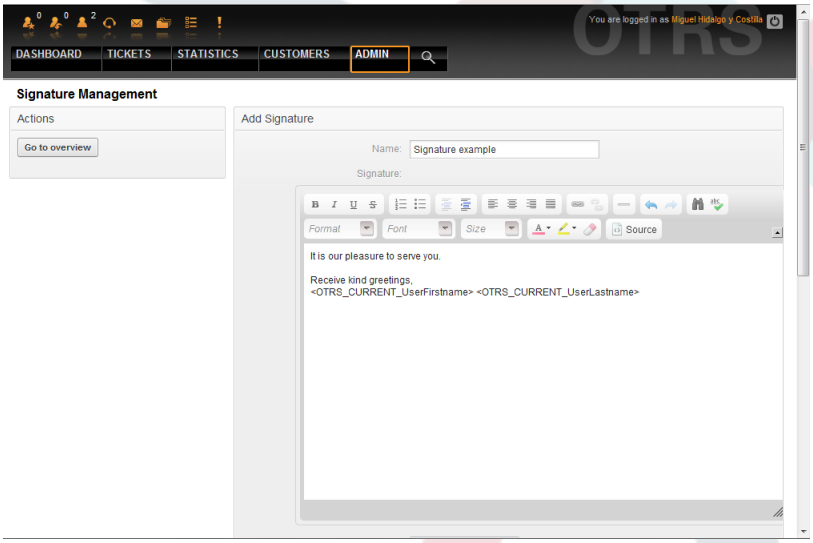
5.2. Signaturen

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.



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.



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>).

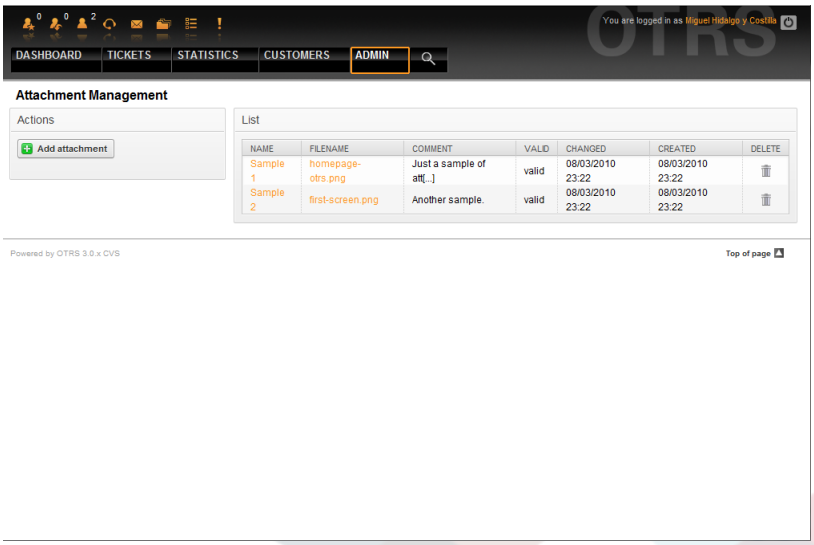
Anmerkung

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.

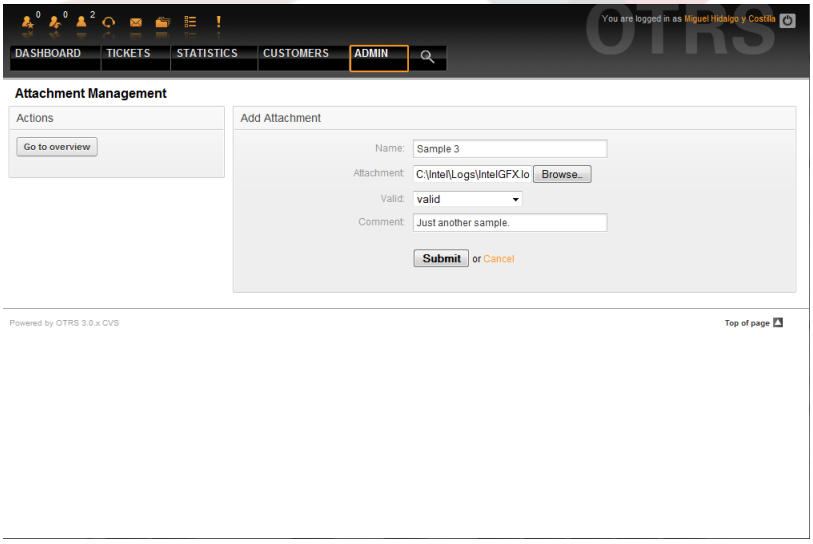
5.3. Anlagen

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.

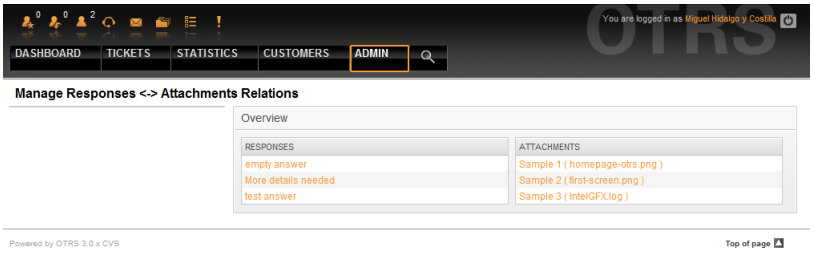
Über den "Anlagen"-Link im Admin-Bereich können Sie neue Anlagen in das System integrieren.



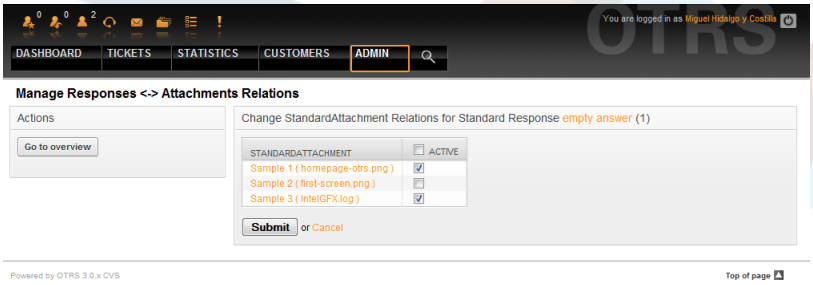
Verwenden Sie den "Anlage hinzufügen"-Knopf, um eine neue Anlage zu erstellen.

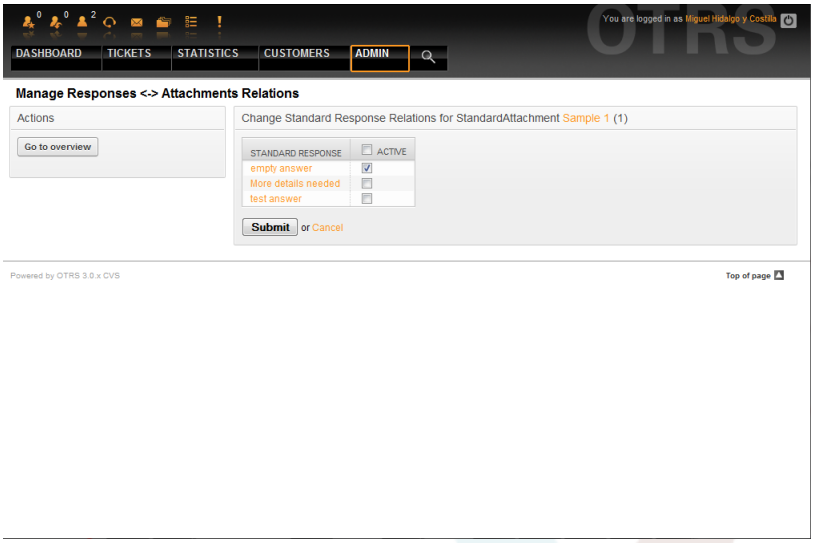


Wenn eine Anlage erstellt wurde, kann sie einer order mehreren Antworten zugeordnet werden. Folgen Sie dazu dem Link "Anlagen <-> Antworten" im Administrationsbereich.



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

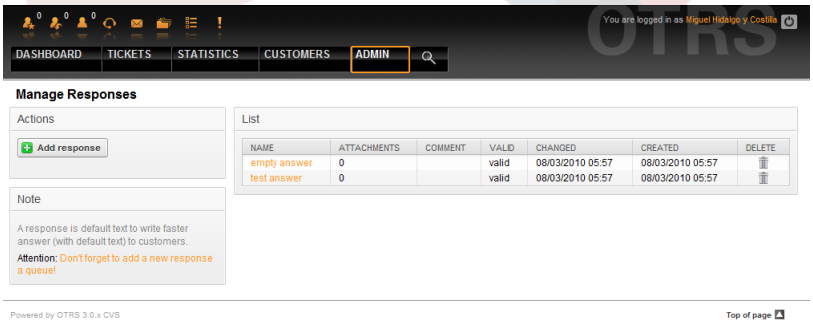




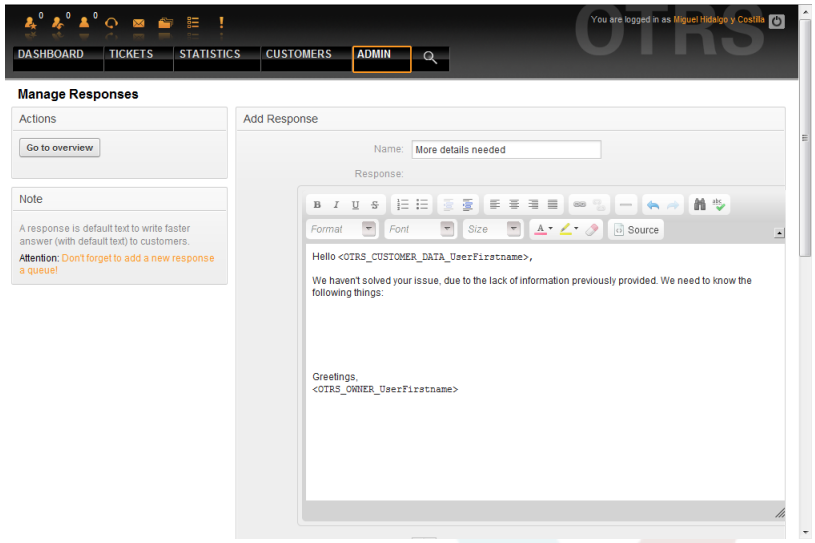
5.4. Antwortvorlagen

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.

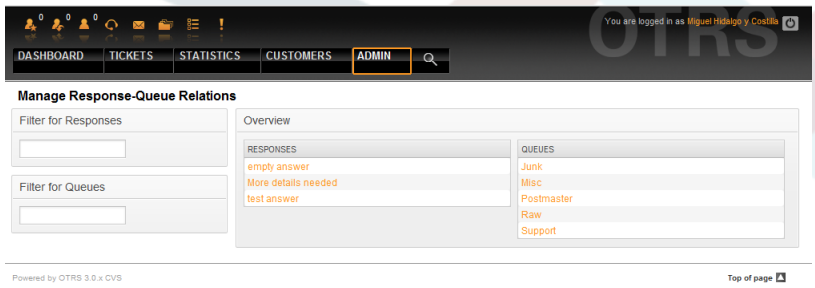
Nach einer Standardinstallation ist für jede Queue die Antwortvorlage "empty answer" vorhanden. Im Admin-Bereich kann über den Link "Antworten" auf die im System gespeicherten Antwortvorlagen zugegriffen und diese bearbeitet werden.



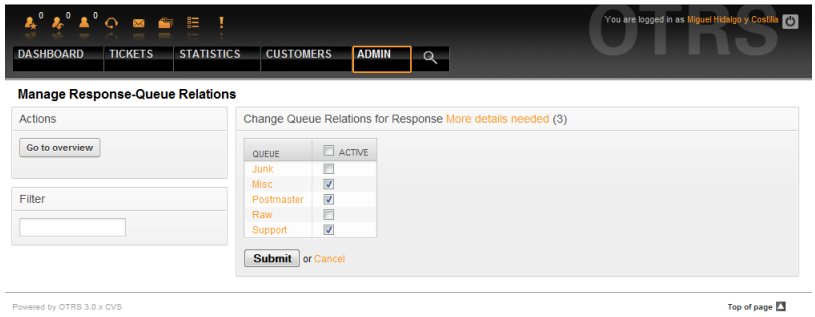
Verwenden Sie den "Antwort hinzufügen"-Knopf zur Erstellung einer neuen Antwortvorlage.

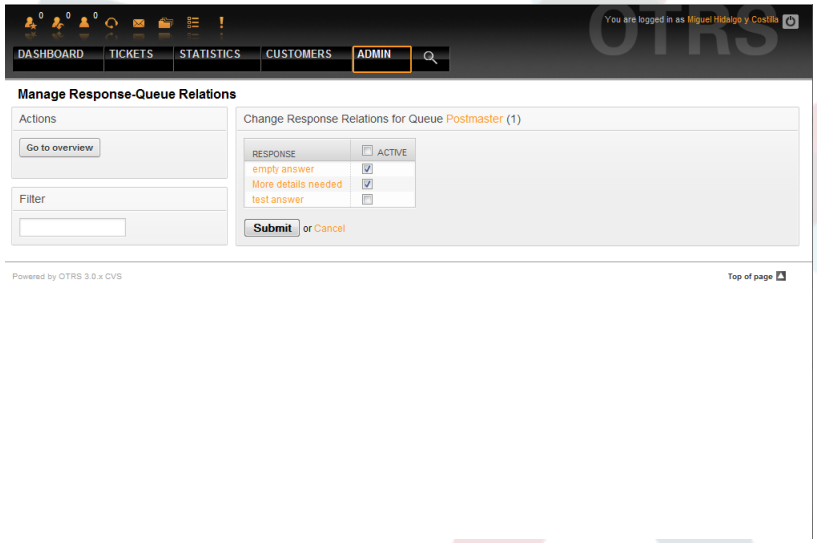


Um eine Antwortvorlage einer oder mehreren Queues zuzuweisen, aktivieren Sie im Admin-Bereich den Link "Antworten <-> Queues". Sie können die Filter verwenden, um bestimmte Elemente zu finden.



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

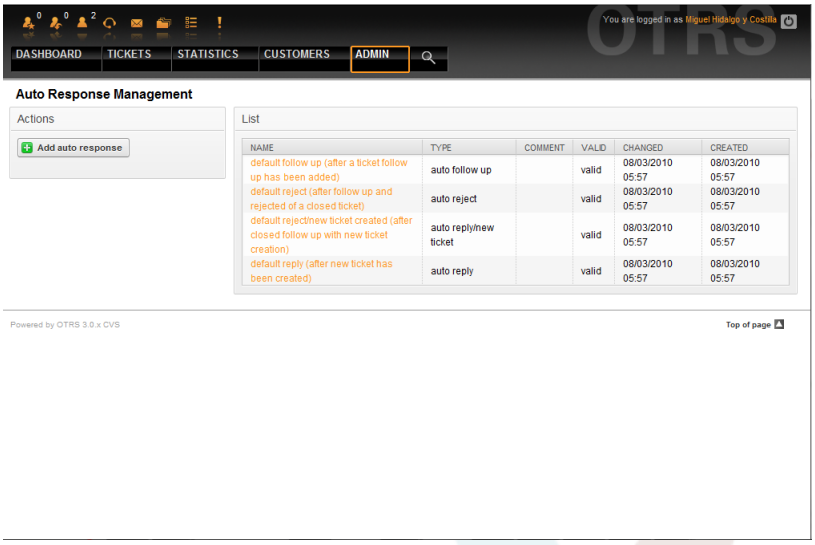




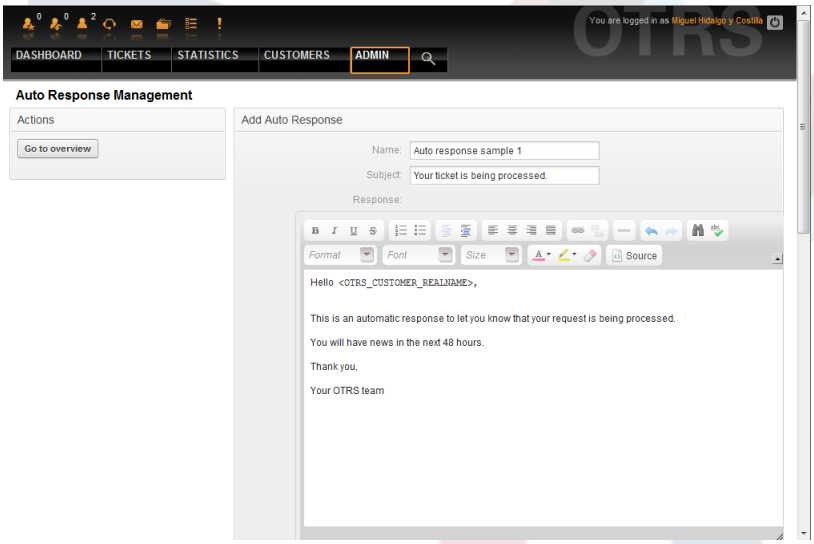
Die Struktur einer Antwort ist intuitiv. Sie enthält eine mit der Queue verknüpfte Anrede, den Text der Antwort, den zitierten Ticket-Text und schließlich die Signatur der Queue.

6. 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.



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



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:

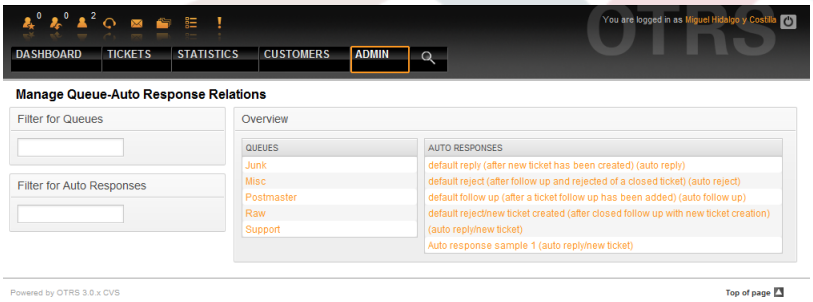
Tabelle 5.3. Ereignisse für automatische Antworten

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.

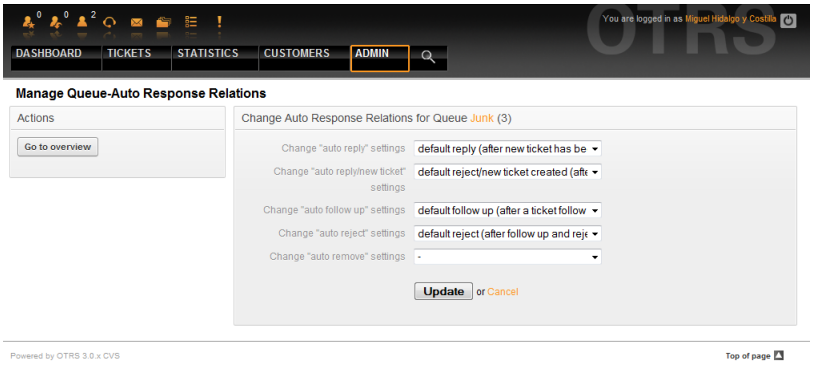
Anmerkung

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.

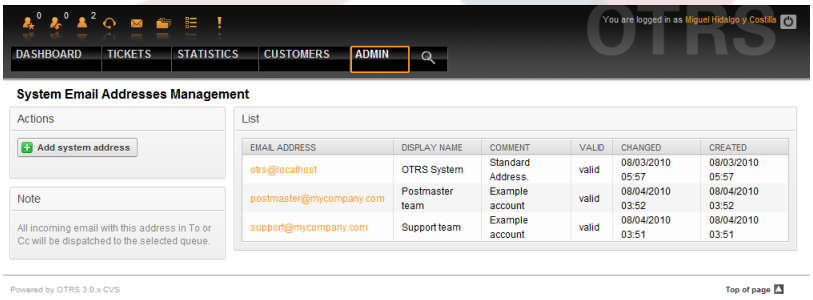


Um die Zuordnungen von automatischen Antworten zu Queues zu verändern, klicken sie auf den Namen der Queue. Durch einen Klick auf den Namen einer automatischen Antwort können Sie diese direkt bearbeiten.

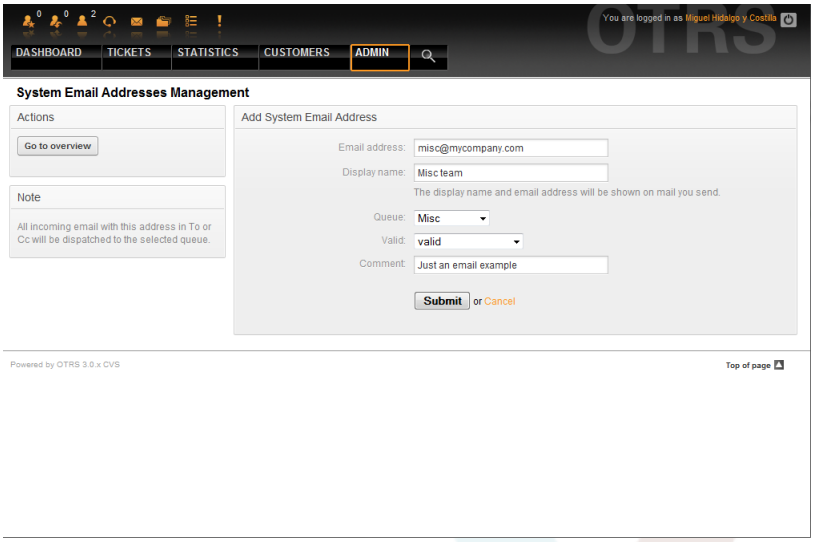


7. E-Mail-Adressen

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.



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.



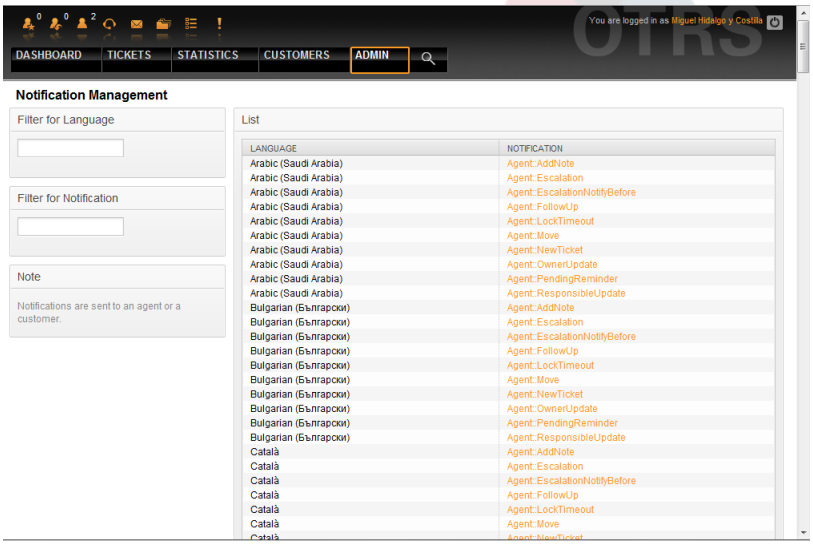
Anmerkung

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.

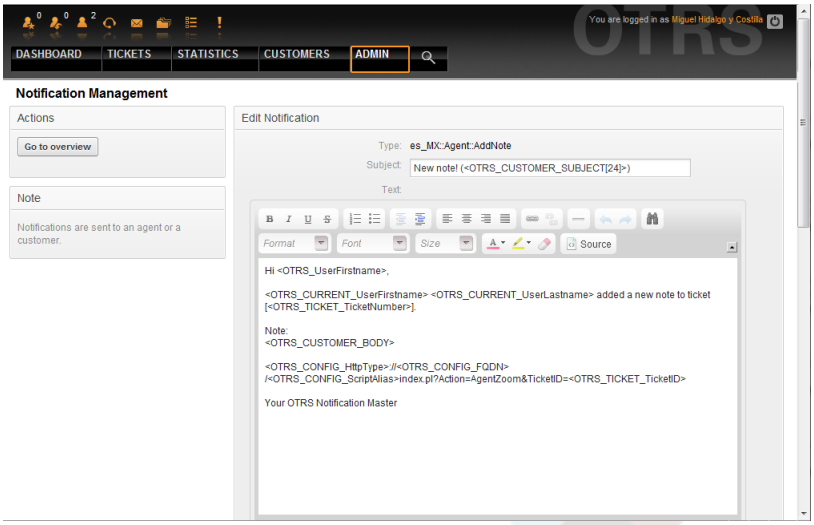
8. Benachrichtigungen

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.



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.

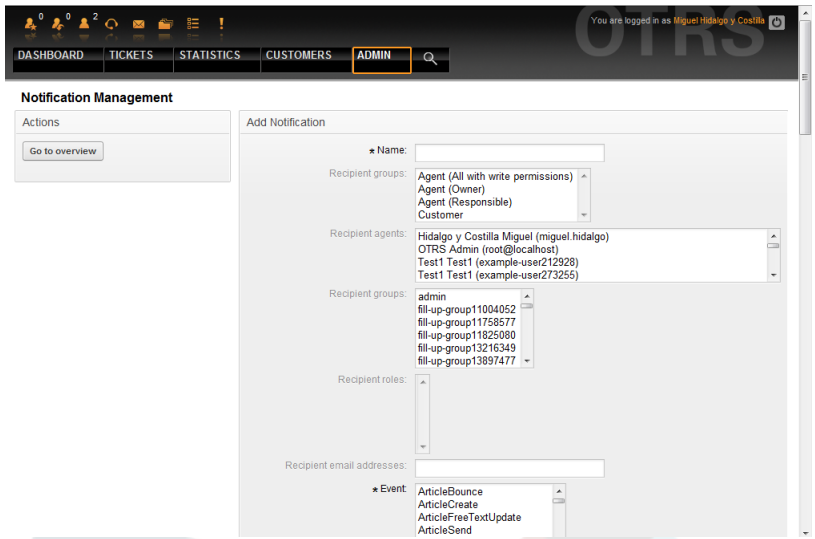


Auch innerhalb der Benachrichtigungen können die Textinhalte mit Hilfe der OTRS-Variablen dynamisch aufbereitet werden. Innerhalb der Anmerkungen im unteren Bereich der Bildschirmmaske zur Benachrichtigungen-Verwaltung werden die verschiedenen zur Verfügung stehenden Variablen und deren Verwendung aufgelistet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

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

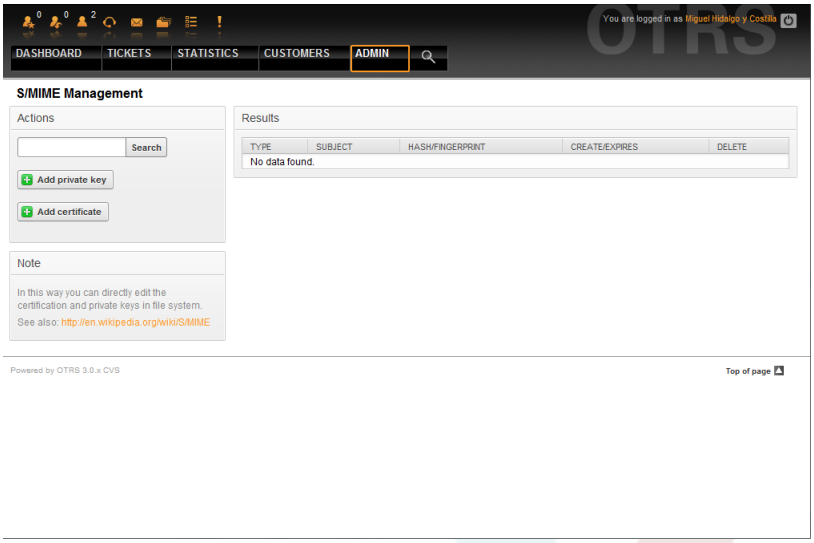


Auch in den ereignisbasierten Benachrichtigungen können Sie mit den speziellen Variablen dynamische Texte erzeugen. Die Referenz der Variablen sehen Sie im unteren Bildschirmbereich.

9. 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.

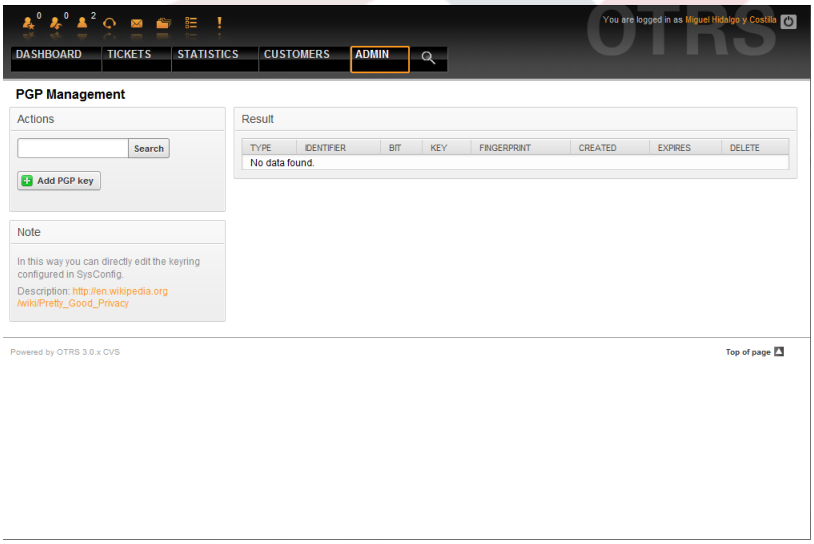
Über den "SMIME"-Link im Admin-Bereich von OTRS erreichen Sie die Verwaltung der SMIME-Zertifikate. Es können Zertifikate und private Fingerprints hinzugefügt und entfernt werden und eine Suche in den Zertifikaten ist möglich.



10. PGP

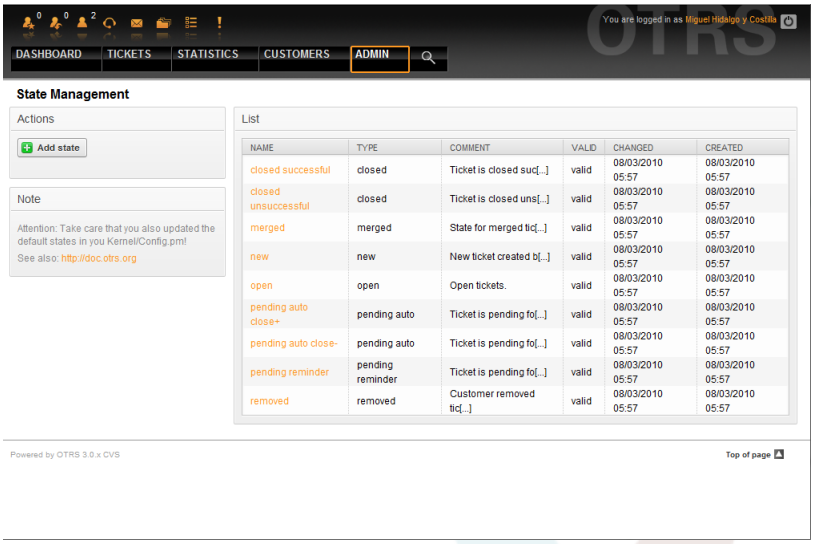
Mit OTRS ist es möglich, Tickets mit Hilfe von PGP zu ent- und verschlüsseln bzw. zu signieren. Bevor PGP allerdings systemweit genutzt werden kann, muss das Feature mit Hilfe einiger Konfigurationsparameter aktiviert und konfiguriert werden.

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.



11. Status

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



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:

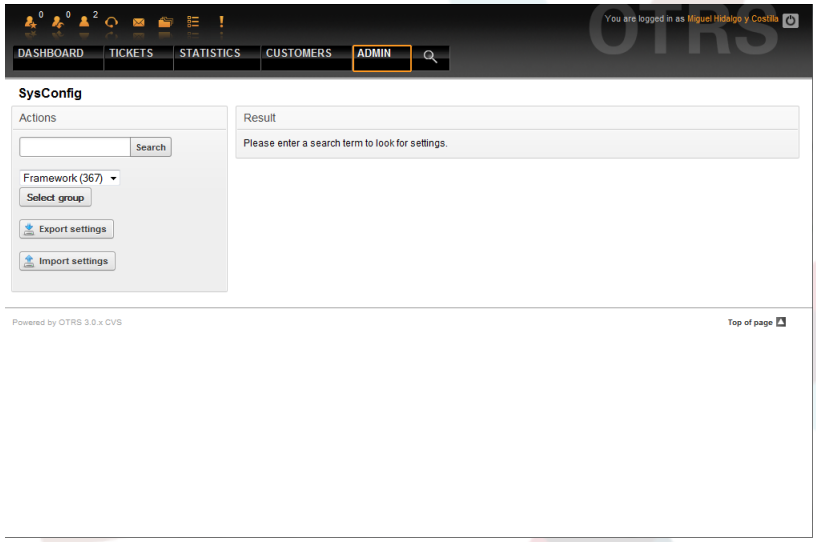
- geschlossen
- zusammengefasst
- neu
- öffnen
- warten auto
- warten zur Erinnerung

- entfernt

12. Die SysConfig (Verwaltung der Systemkonfiguration)

In der SysConfig werden die meisten Konfigurationseinstellungen von OTRS vorgenommen.

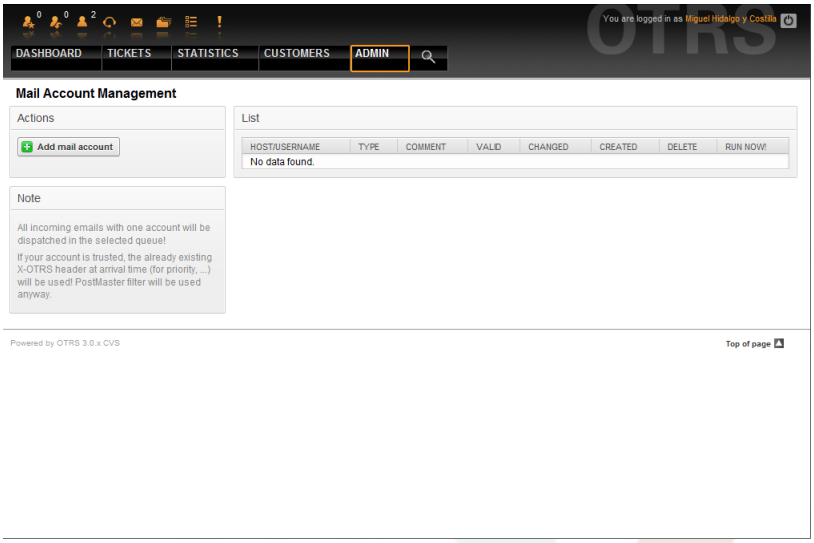
Über den "SysConfig"-Link im Admin-Bereich von OTRS erreichen sie die grafische Administrationsoberfläche. Über diese können Sie eigene Konfigurationsdateien in das System integrieren bzw. Ihre persönlichen Änderungen an der Standardkonfiguration in eine Datei sichern. Nahezu alle Konfigurationsparameter 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.



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

13. Einrichten von Mail-Konten

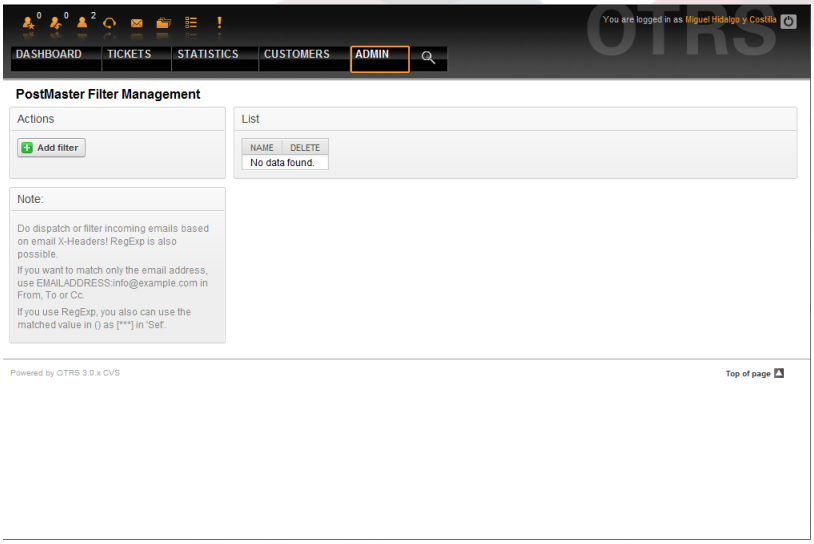
There are several possibilities to transport new emails into the ticket system. One way is to use a local MTA and the otrs.PostMaster.pl script that pipes the mails directly into the system. Another possibility is the use of mail accounts which can be administrated through the web interface. The "PostMaster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



See the section about PostMaster Mail Accounts for more details.

14. Filtering incoming email messages

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



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

ticket or follow-up message in OTRS. The Table 5-4 lists the different X-OTRS headers and their meaning.

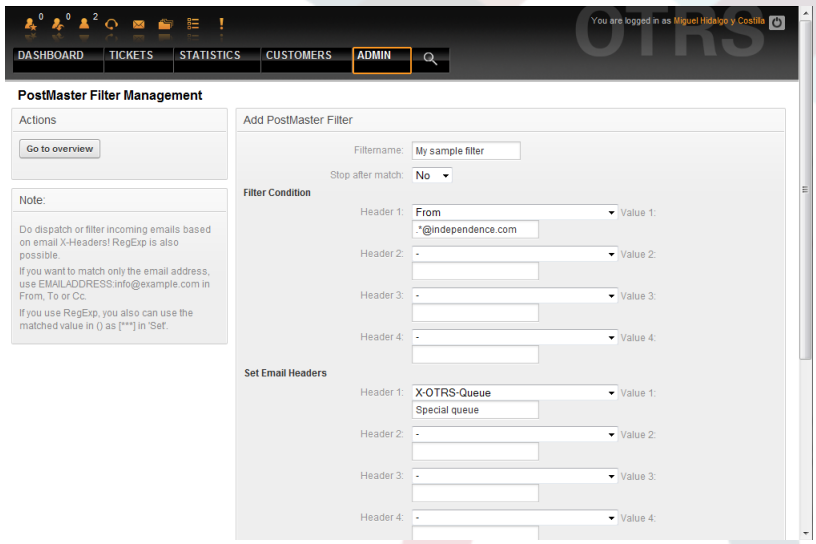
Note: You also can use X-OTRS-FollowUp-* headers to set values for follow up emails.

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.	Sets the queue where the ticket shall be sorted. If set in X-OTRS header, all other filter rules that try to sort a ticket into a specific queue are ignored. If you use a sub-queue, specify it as "Parent::Sub".
X-OTRS-Lock:	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)	Sets the type of a ticket (if Ticket::Type is activated).
X-OTRS-Service:	(systemabhängig)	Sets the service of a ticket (if Ticket::Service is active). If you want to set a sub-service you should specify it as "Parent::Sub".
X-OTRS-SLA:	(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.

Name	Mögliche Werte	Beschreibung
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.

You should specify a name for every filter rule. Filter criteria can be specified in the section "Filter Condition". Choose via the listboxes for "Header 1", "Header 2" and so on for the parts of the messages where you would like to search, and specify on the right side the values you wish to filter on. In the section "Set Email Headers", you can choose the actions that are triggered if the filter rules match. You can select for "Header 1", "Header 2" and so on to select the X-OTRS-Header and set the associated values (see Figure below).



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".

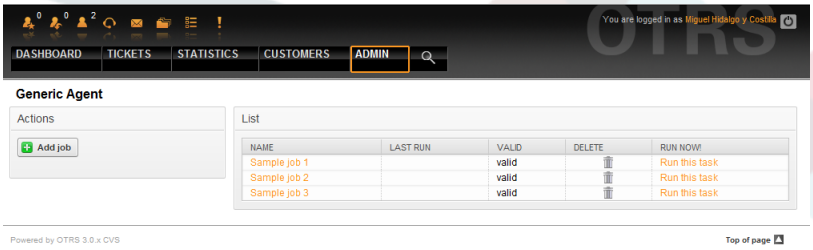
Um nun eine Filterregel mit diesem Kriterium zu erzeugen, tragen Sie hierzu als Name für die Filterregel z. B. "spam-mails" ein und wählen in der Sektion "Treffer" für "Header 1" den Header "X-Spam-Flag:" aus der Listbox aus. Rechts daneben fügen Sie als Wert "Yes" hinzu. Das Filterkriterium wurde somit 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-Mailsgenutzt werden und bei komplexeren Installationen nützlich sein können.

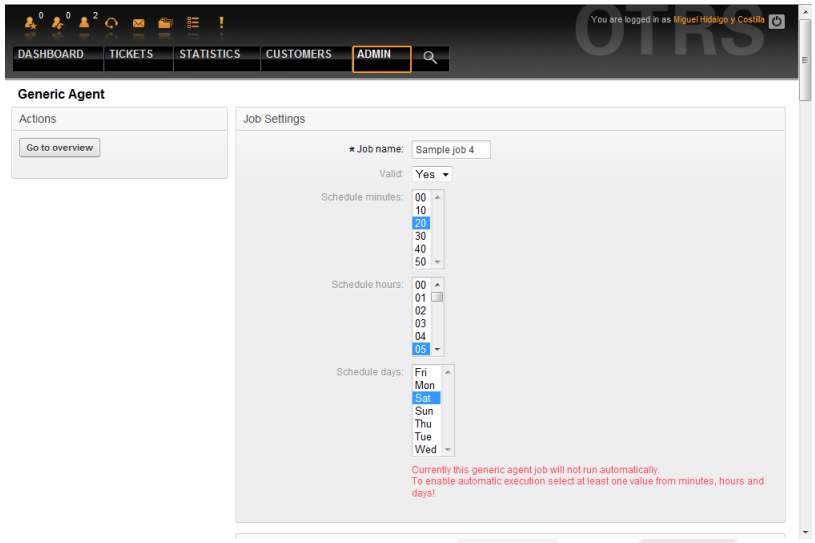
15. Ausführen von automatisierten Jobs mit Hilfe des GenericAgents

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.



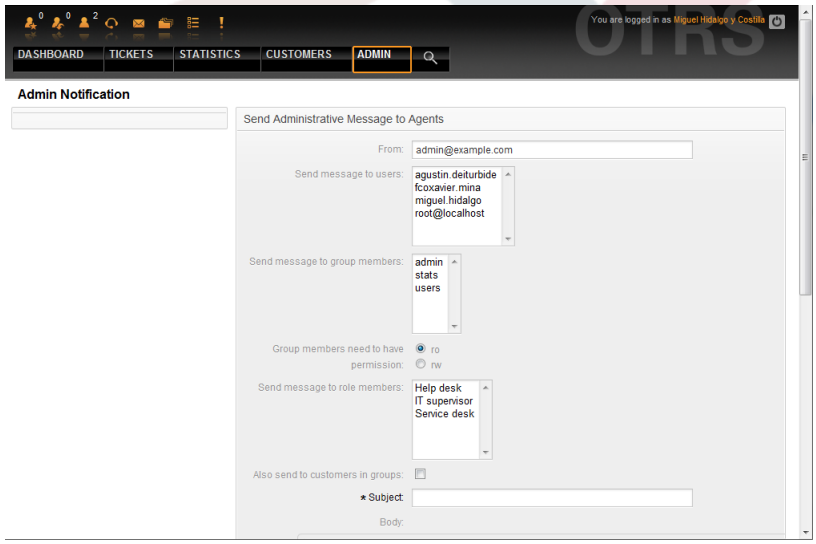
Über die Bildschirmmaske zum Erstellen eines neuen Jobs kann der Zeitplan für die Ausführung des Jobs eingestellt werden. Desweiteren kann über verschiedene Kriterien festgelegt werden, welche Tickets vom Job erfasst werden sollen. Schließlich ist es möglich, die neuen Eigenschaften der vom Job betroffenen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



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.

16. Administrator-E-Mail

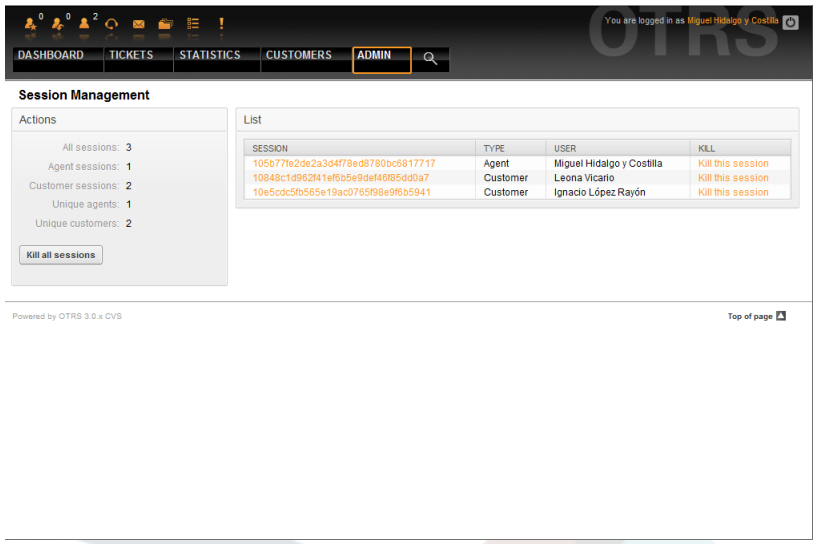
Um als OTRS-Administrator eine Mail an bestimmte Benutzer (Agenten) oder Benutzergruppen im System zu versenden, folgen Sie dem Link "Admin Notification".



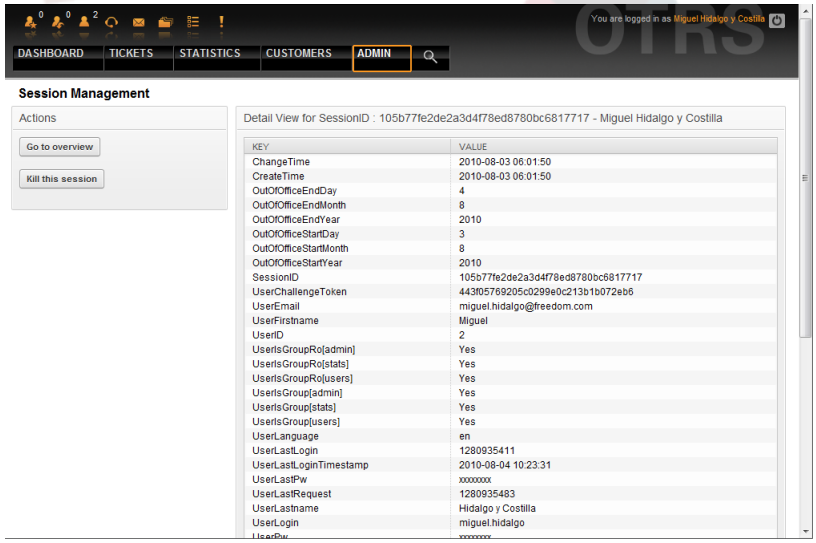
Es wird ein Formular geladen, in das Sie die Absenderadresse, den Betreff und den Inhalt der Administratoren-Mitteilung eintragen können. Weiterhin können Sie aus der Tabelle auswählen, an welche Benutzer und / oder Benutzergruppen bzw. Rollen die Mitteilung gesendet werden soll.

17. Sitzungsverwaltung

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

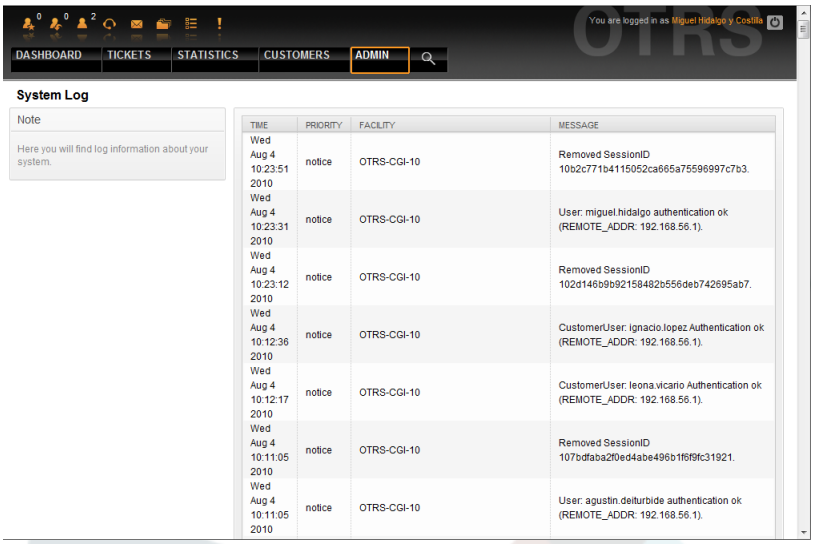


Innerhalb der Sitzungsverwaltung werden allgemeine Informationen zu allen Sitzungen im System ausgegeben, also z. B. wie viele Sitzungen gerade insgesamt aktiv sind oder wie viele Agenten- und Kunden-Sitzungen laufen, usw. Es besteht die Möglichkeit mit Hilfe des "Alle Sitzungen löschen"-Schalters die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



18. System Log

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



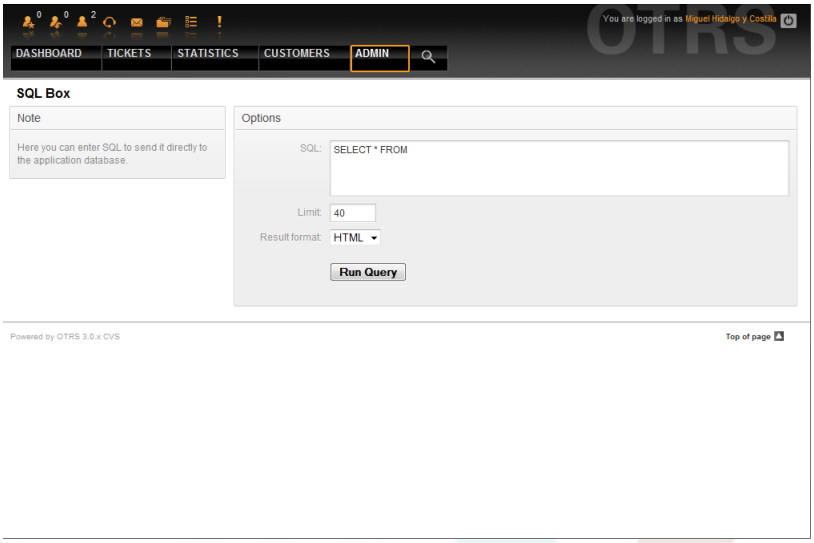
Ein Logeintrag setzt sich aus der Zeit, der Priorität, der betroffenen Systemkomponente und der eigentlichen Meldung zusammen.

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.

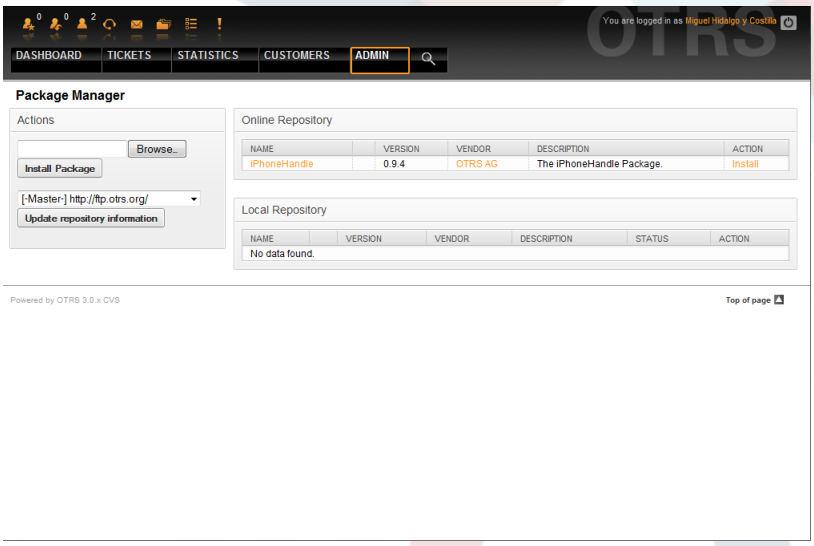
19. SQL-Abfragen mit Hilfe der SQL Box

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.



20. 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.



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 auswählen und auf klicken. Die verfügbaren Pakete werden rechts tabellarisch dargestellt. Um eines zu installieren, klicken Sie auf "Installieren". Nach der Installation wird das Paket im Abschnitt dargestellt.

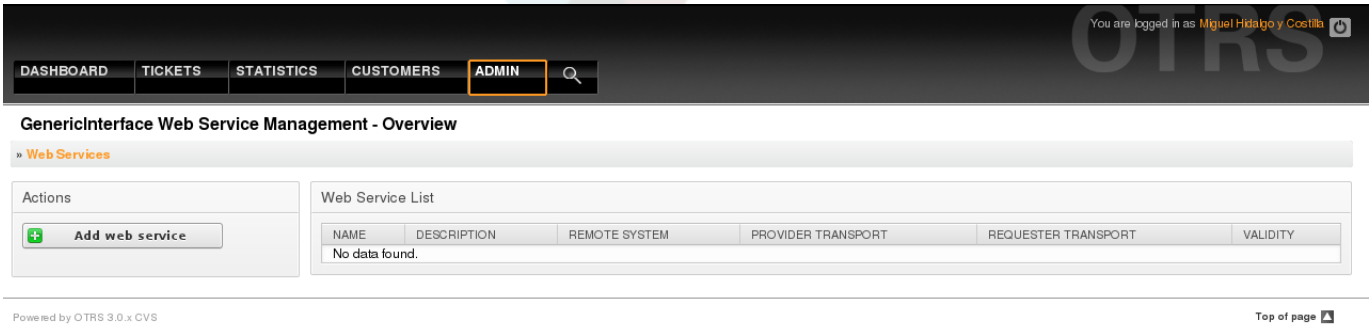
Um ein installiertes Paket zu aktualisieren, können Sie die Funktion "Aktualisieren" benutzen, die bei installierten Paketen angezeigt wird, von denen eine neuere Version verfügbar ist. Klicken Sie einfach auf "Aktualisieren", und die neue Version wird in Ihrem System installiert.

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 ⁓⁓⁓⁓⁓ ⁓⁓⁓⁓⁓⁓⁓⁓⁓-Knopf in der "Aktionen"-Liste, und wählen Sie die .opm-Datei auf Ihrem Speichermedium aus. Verwenden Sie dann ⁓⁓⁓⁓⁓ ⁓⁓⁓⁓⁓⁓⁓⁓⁓⁓⁓. Nach der Installation wird das Paket im Abschnitt ⁓⁓⁓⁓⁓⁓⁓ ⁓⁓⁓⁓⁓⁓⁓⁓⁓⁓⁓ 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.

21. 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).

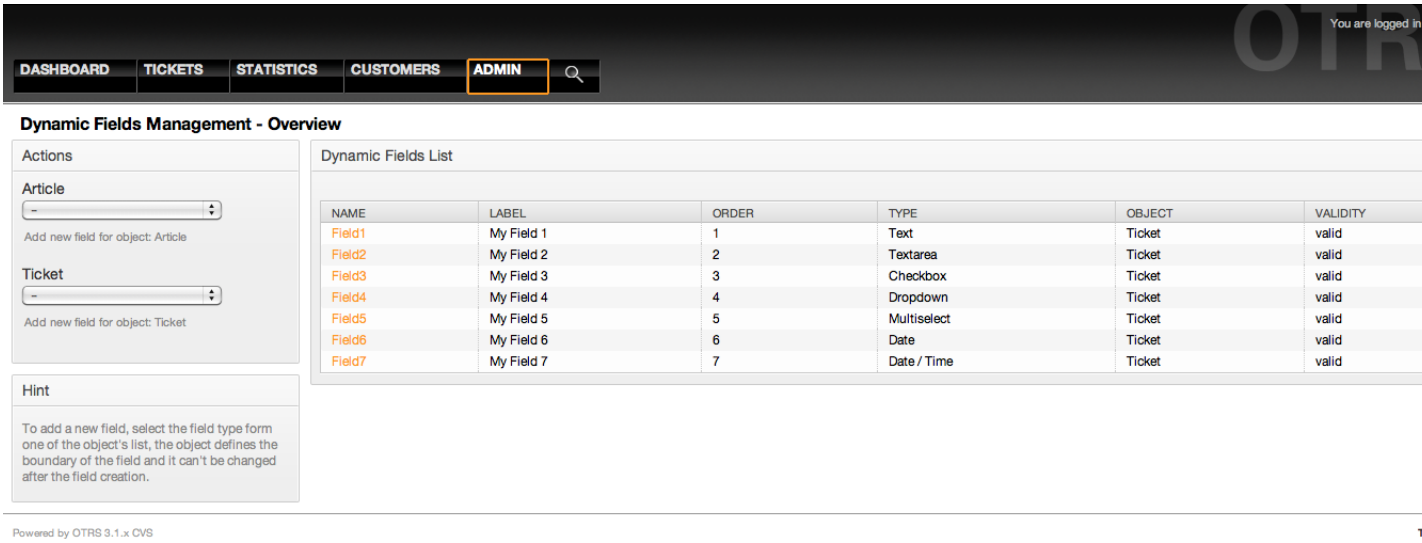


⁓⁓⁓⁓⁓⁓⁓ ⁓⁓⁓ ⁓⁓⁓⁓⁓⁓⁓⁓⁓ ⁓⁓⁓⁓⁓⁓⁓⁓⁓ ⁓⁓⁓ ⁓⁓⁓ ⁓⁓⁓⁓⁓⁓⁓⁓⁓⁓

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

22. Dynamic Fields

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



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

1. 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.

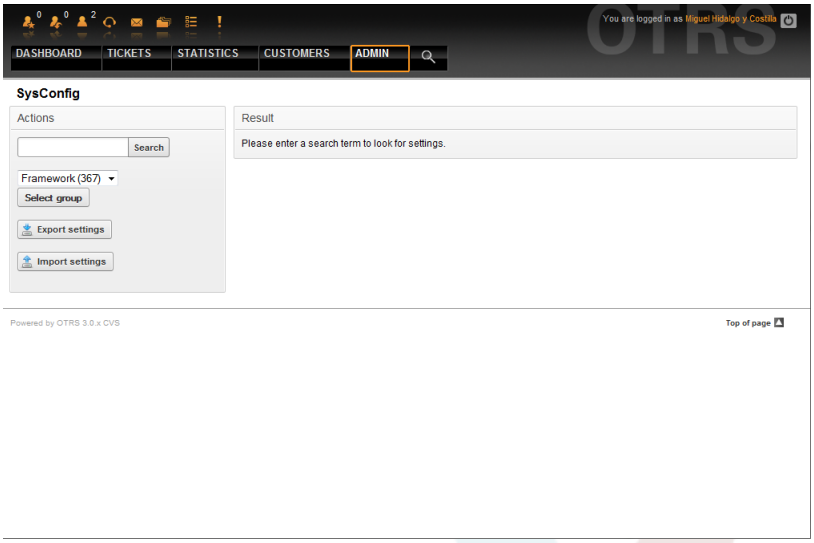
Die Datei `Kernel/Config/Defaults.pm` enthält die Konfigurationsparameter für den OTRS-Framework. In ihr finden Sie grundlegende Einstellungen wie die Mailkonfiguration, die Datenbankanbindung, Standardsprache o.ä. In der Datei`Kernel/Config/Files/Ticket.pm` sind alle Konfigurationsparameter für das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

2. 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).



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 Datenbankanbindung 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

1. E-Mails versenden

1.1. Via Sendmail (Standard)

OTRS ist in der Lage, E-Mails via Sendmail (z. B. Sendmail, Postfix, Qmail oder Exim) zu versenden. Die Standard-Konfiguration sollte gleich ohne Probleme funktionieren.

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

1.2. Via SMTP server/smarthost

Wenn kein sendmail-Binary zur Verfügung steht, kann OTRS E-Mails via SMTP (Simple Mail Transfer Protocol / RFC 821) 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.

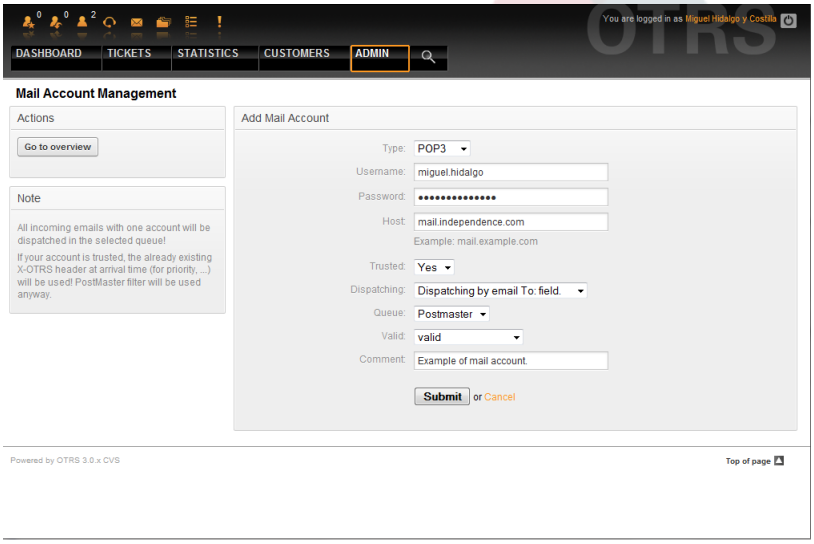
2. E-Mails empfangen

2.1. 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.



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 abgearbietet, 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.

2.2. 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#
```

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 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
```

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

2.3. 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.

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.

2.4. 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

1. 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.

1.1. 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 Erinnerungs-Tickets 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.

1.2. 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 ~~~~~~~~~~ Feiertage voreingestellt.

1.3. 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.

2. 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 [illegible] definiert.
- Das Ticket ist [illegible].
- Der Status des Tickets is [illegible].

Die Freigabezeit wird zurückgesetzt, wenn ein Agent einen neuen externen Artikel zum Ticket hinzufügt. Das kann einer der folgenden Artikeltypen sein: [illegible], [illegible], [illegible], [illegible], or [illegible].

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

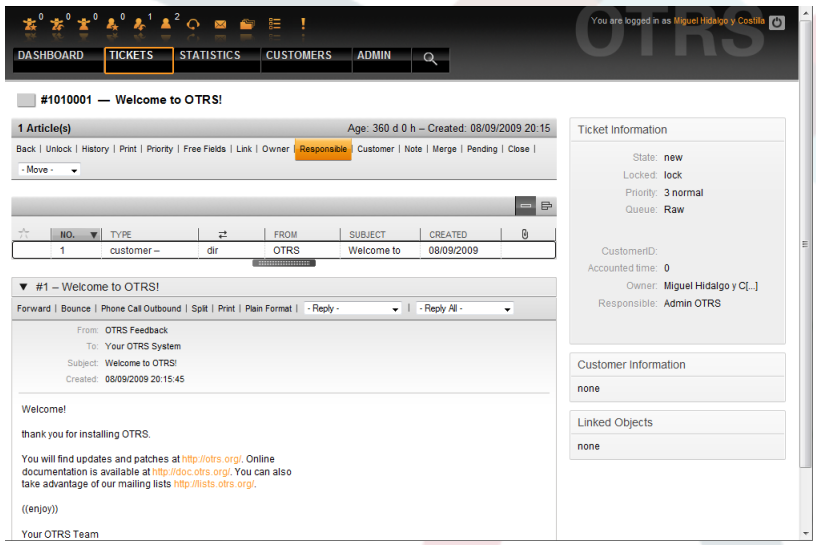
Seit OTRS 2.1 kann zusätzlich zum Besitzer ein Verantwortlicher für das Ticket festgelegt werden. Zudem können alle Aktivitäten rund um das Ticket beobachtet werden. Diese beiden Funktionalitäten sind mit den TicketResponsible- und dem TicketWatcher-Funktionen implementiert, und erleichtern das Zuweisen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

1. Ticket-Verantwortlicher

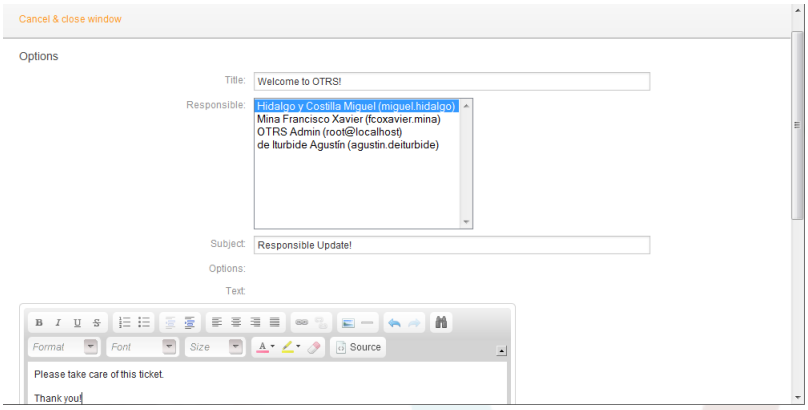
Die Funktion "Verantwortlicher" eines Tickets macht es möglich, das Ticket einem anderen als dem aktuellen Besitzer zur Bearbeitung zu übergeben. Ein Benutzer, der ein Ticket gesperrt hat, kann es einem anderen Agenten, der nicht der Ticketbesitzer ist, zur Beantwortung übergeben. Nach der Bearbeitung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



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.



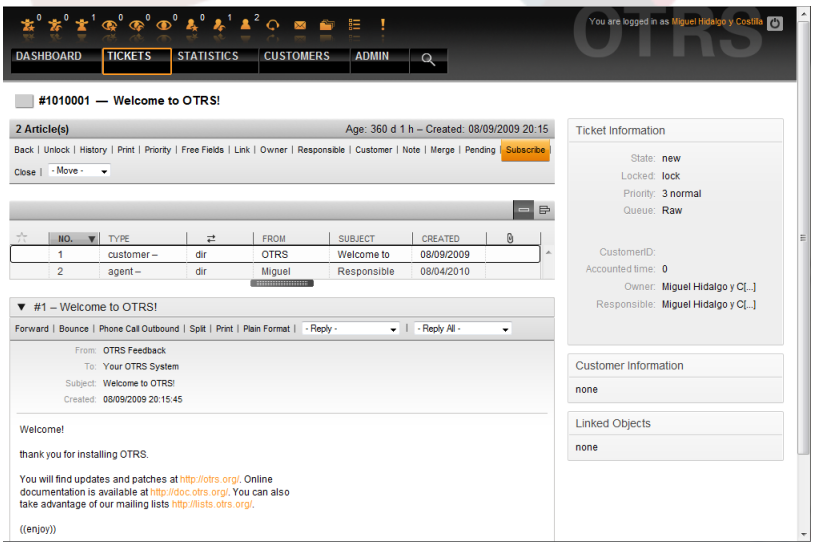
Die Liste aller Tickets, für die ein Benutzer verantwortlich ist, kann durch die Ansicht "Verantwortliche Tickets" eingesehen werden. Diese kann über 3 Icons in der Toolbar aufgerufen werden, sobald das TicketResponsible-Feature aktiviert wird.

2. Tickets beobachten

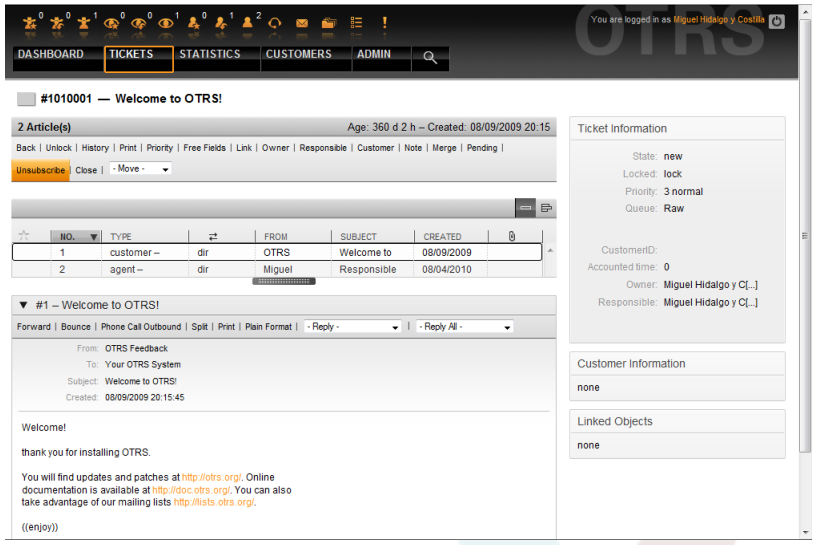
Sollen innerhalb des Ticket-Systems lediglich bestimmte Tickets z. B. von einem Abteilungsleiter beobachtet und nicht selbst bearbeitet werden, so kann dies seit OTRS 2.1 mit Hilfe des TicketWatcher-Features realisiert werden.

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.

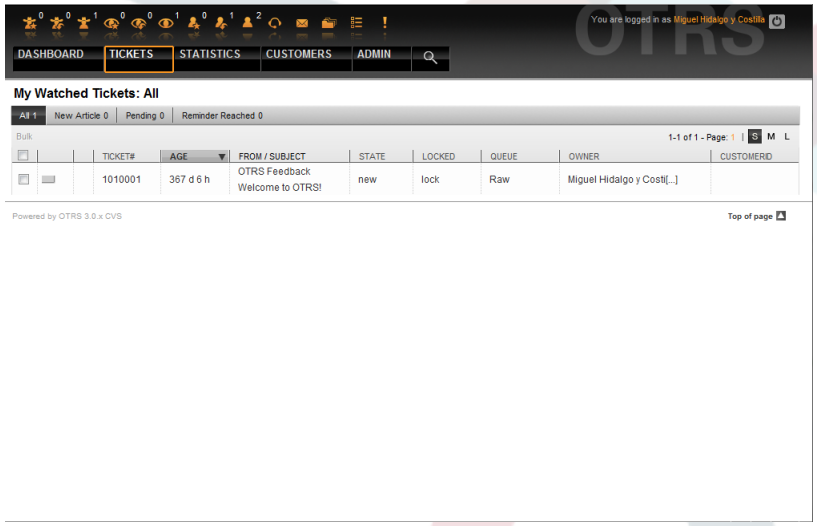
Um ein Ticket zu beobachten, muss der Inhalt des Tickets aufgerufen und im Menü für die verschiedenen Aktionen für das Ticket der "Beobachten"-Schalter betätigt werden.



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



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.



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

1. 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;
```

2. 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.

2.1. 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',
#            Type => 'mssql', # only for ODBC connections
#            DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',
#            User => '',
#            Password => '',
            Table => 'customer_user',
            # if your frontend is unicode and the charset of your
            # customer database server is iso-8859-1, use these options.
#           SourceCharset => 'iso-8859-1',
#           DestCharset => 'utf-8',
```

```
            # CaseSensitive will control if the SQL statements need LOWER()
            #   function calls to work case insensitively. Setting this to
            #   1 will improve performance dramatically on large databases.
            CaseSensitive => 0,
        },
# customer unique id
CustomerKey => 'login',

# customer #
CustomerID => 'customer_id',
CustomerValid => 'valid_id',
    CustomerUserListFields => ['first_name', 'last_name', 'email'],
    CustomerUserSearchFields => ['login', 'last_name', 'customer_id'],
    CustomerUserSearchPrefix => '',
    CustomerUserSearchSuffix => '*',
    CustomerUserSearchListLimit => 250,
    CustomerUserPostMasterSearchFields => ['email'],
    CustomerUserNameFields => ['title','first_name','last_name'],
    CustomerUserEmailUniqCheck => 1,
#    # show not own tickets in customer panel, CompanyTickets
#    CustomerUserExcludePrimaryCustomerID => 0,
#    # generate auto logins
#    AutoLoginCreation => 0,
#    AutoLoginCreationPrefix => 'auto',
#    # admin can change customer preferences
#    AdminSetPreferences => 1,
#    # cache time to live in sec. - cache any database queries
#    CacheTTL => 0,
#    # just a read only source
#    ReadOnly => 1,
    Map => [
        # note: Login, Email and CustomerID needed!
        # var, frontend, storage, shown (1=always,2=lite), required, storage-type, http-link,
 readonly, http-link-target
        [ 'UserTitle',      'Title',      'title',      1, 0, 'var', '', 0 ],
        [ 'UserFirstname',  'Firstname',  'first_name', 1, 1, 'var', '', 0 ],
        [ 'UserLastname',   'Lastname',   'last_name',  1, 1, 'var', '', 0 ],
        [ 'UserLogin',      'Username',   'login',      1, 1, 'var', '', 0 ],
        [ 'UserPassword',   'Password',   'pw',         0, 0, 'var', '', 0 ],
        [ 'UserEmail',      'Email',      'email',      1, 1, 'var', '', 0 ],

#        [ 'UserEmail',      'Email', 'email',           1, 1, 'var', '$Env{"CGIHandle"}?
Action=AgentTicketCompose&ResponseID=1&TicketID=$Data{"TicketID"}&ArticleID=
$Data{"ArticleID"}', 0 ],
        [ 'UserCustomerID', 'CustomerID', 'customer_id', 0, 1, 'var', '', 0 ],

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

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:~#
```

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 ],
```

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

2.1.1. 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:~#
```

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 ],
```

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.

2.2. 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 ],
```

2.2.1. 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 ],
```

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.

2.3. 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',
#        Type => 'mssql', # only for ODBC connections
#        DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',
#        User => '',
#        Password => '',
        Table => 'customer_user',
    },
    # customer unique id
    CustomerKey = 'login',
    # customer #
    CustomerID = 'customer_id',
    CustomerValid = 'valid_id',
    CustomerUserListFields => ['first_name', 'last_name', 'email'],
    CustomerUserSearchFields => ['login', 'last_name', 'customer_id'],
    CustomerUserSearchPrefix => '',
    CustomerUserSearchSuffix => '*',
    CustomerUserSearchListLimit => 250,
    CustomerUserPostMasterSearchFields => ['email'],
    CustomerUserNameFields => ['title','first_name','last_name'],
    CustomerUserEmailUniqCheck => 1,
#    # show not own tickets in customer panel, CompanyTickets
#    CustomerUserExcludePrimaryCustomerID => 0,
#    # generate auto logins
#    AutoLoginCreation => 0,
#    AutoLoginCreationPrefix => 'auto',
#    # admin can change customer preferences
#    AdminSetPreferences => 1,
#    # cache time to live in sec. - cache any database queries
#    CacheTTL => 0,
#    # just a read only source
#    ReadOnly => 1,
    Map => [

        # note: Login, Email and CustomerID needed!
        # var, frontend, storage, shown (1=always,2=lite), required, storage-type, http-link,
 readonly, http-link-target
        [ 'UserTitle',      'Title',      'title',        1, 0, 'var', '', 0 ],
        [ 'UserFirstname',  'Firstname',  'first_name',   1, 1, 'var', '', 0 ],
        [ 'UserLastname',   'Lastname',   'last_name',    1, 1, 'var', '', 0 ],
        [ 'UserLogin',      'Username',   'login',        1, 1, 'var', '', 0 ],
        [ 'UserPassword',   'Password',   'pw',           0, 0, 'var', '', 0 ],
        [ 'UserEmail',      'Email',      'email',        1, 1, 'var', '', 0 ],
        [ 'UserCustomerID', 'CustomerID', 'customer_id',  0, 1, 'var', '', 0 ],
        [ 'UserPhone',      'Phone',      'phone',        1, 0, 'var', '', 0 ],
        [ 'UserFax',        'Fax',        'fax',          1, 0, 'var', '', 0 ],
        [ 'UserMobile',     'Mobile',     'mobile',       1, 0, 'var', '', 0 ],
        [ 'UserStreet',     'Street',     'street',       1, 0, 'var', '', 0 ],
        [ 'UserZip',        'Zip',        'zip',          1, 0, 'var', '', 0 ],
        [ 'UserCity',       'City',       'city',         1, 0, 'var', '', 0 ],
        [ 'UserCountry',    'Country',    'country',      1, 0, 'var', '', 0 ],
        [ 'UserComment',    'Comment',    'comments',     1, 0, 'var', '', 0 ],
        [ 'ValidID',        'Valid',      'valid_id',     0, 1, 'int', '', 0 ],
    ],
    # default selections
    Selections => {
        UserTitle => {
            'Mr.' => 'Mr.',
            'Mrs.' => 'Mrs.',
        },
    },
};

# 2. Customer user backend: LDAP
```

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

        # Net::LDAP new params (if needed - for more info see perldoc Net::LDAP)
        Params => {
            port => 389,
            timeout => 120,
            async => 0,
            version => 3,
        },
    },
    # customer unique id
    CustomerKey => 'uid',
    # customer #
    CustomerID => 'mail',
    CustomerUserListFields => ['cn', 'mail'],
    CustomerUserSearchFields => ['uid', 'cn', 'mail'],
    CustomerUserSearchPrefix => '',
    CustomerUserSearchSuffix => '*',
    CustomerUserSearchListLimit => 250,
    CustomerUserPostMasterSearchFields => ['mail'],
    CustomerUserNameFields => ['givenname', 'sn'],
    # show not own tickets in customer panel, CompanyTickets
    CustomerUserExcludePrimaryCustomerID => 0,
    # add a ldap filter for valid users (expert setting)
#    CustomerUserValidFilter => '(!(description=locked))',
    # admin can't change customer preferences
    AdminSetPreferences => 0,
    Map => [
        # note: Login, Email and CustomerID needed!
        # var, frontend, storage, shown (1=always,2=lite), required, storage-type, http-link,
 readonly
        [ 'UserTitle',      'Title',      'title',           1, 0, 'var', '', 0 ],
        [ 'UserFirstname',  'Firstname',  'givenname',       1, 1, 'var', '', 0 ],
        [ 'UserLastname',   'Lastname',   'sn',              1, 1, 'var', '', 0 ],
        [ 'UserLogin',      'Username',   'uid',             1, 1, 'var', '', 0 ],
        [ 'UserEmail',      'Email',      'mail',            1, 1, 'var', '', 0 ],
        [ 'UserCustomerID', 'CustomerID', 'mail',            0, 1, 'var', '', 0 ],
#        [ 'UserCustomerIDs', 'CustomerIDs', 'second_customer_ids', 1, 0, 'var', '', 0 ],
        [ 'UserPhone',      'Phone',      'telephonenumber', 1, 0, 'var', '', 0 ],
        [ 'UserAddress',    'Address',    'postaladdress',   1, 0, 'var', '', 0 ],
        [ 'UserComment',    'Comment',    'description',     1, 0, 'var', '', 0 ],
    ],
};
```

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.

3. Backends für die Authentifizierung von Agenten und Kunden

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

3.1. Authentifizierungs-Backends für Agenten

3.1.1. 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';
```

3.1.2. 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',
];
```

3.1.3. 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 singe login through
# apache http-basic-auth
$Self->{'AuthModule'} = 'Kernel::System::Auth::HTTPBasicAuth';
```

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

3.1.4. 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';
```

3.2. Authentifizierungs-Backends für Kunden

3.2.1. 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 againt 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";
```

3.2.2. 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,
};
```

3.2.3. 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';
```

3.2.4. 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';
```

4. Customizing the customer self-registration

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.

4.1. 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>
[...]
```

4.2. 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',
#        Type => 'mssql', # only for ODBC connections
#        DSN => 'DBI:mysql:database=customerdb;host=customerdbhost',
#        User => '',
#        Password => '',
        Table => 'customer_user',
    },
    # customer unique id
    CustomerKey => 'login',
    # customer #
    CustomerID => 'customer_id',
    CustomerValid => 'valid_id',
    CustomerUserListFields => ['first_name', 'last_name', 'email'],
#    CustomerUserListFields => ['login', 'first_name', 'last_name', 'customer_id', 'email'],
    CustomerUserSearchFields => ['login', 'last_name', 'customer_id'],
    CustomerUserSearchPrefix => '',
    CustomerUserSearchSuffix => '*',
    CustomerUserSearchListLimit => 250,
    CustomerUserPostMasterSearchFields => ['email'],
    CustomerUserNameFields => ['title', 'first_name', 'last_name'],
```

```
    CustomerUserEmailUniqCheck => 1,
#    # show not own tickets in customer panel, CompanyTickets
#    CustomerUserExcludePrimaryCustomerID => 0,
#    # generate auto logins
#    AutoLoginCreation => 0,
#    AutoLoginCreationPrefix => 'auto',
#    # admin can change customer preferences
#    AdminSetPreferences => 1,
#    # cache time to live in sec. - cache database queries
#    CacheTTL => 0,
#    # just a read only source
#    ReadOnly => 1,
    Map => [

        # note: Login, Email and CustomerID needed!
        # var, frontend, storage, shown (1=always,2=lite), required, storage-type, http-link,
 readonly, http-link-target
        [ 'UserTitle',      'Title',      'title',       1, 0, 'var', '', 0 ],
        [ 'UserFirstname',  'Firstname',  'first_name',  1, 1, 'var', '', 0 ],
        [ 'UserLastname',   'Lastname',   'last_name',   1, 1, 'var', '', 0 ],
        [ 'UserLogin',      'Username',   'login',       1, 1, 'var', '', 0 ],
        [ 'UserPassword',   'Password',   'pw',          0, 0, 'var', '', 0 ],
        [ 'UserEmail',      'Email',      'email',       1, 1, 'var', '', 0 ],
        [ 'UserCustomerID', 'CustomerID', 'customer_id', 0, 1, 'var', '', 0 ],
        [ 'UserPhone',      'Phone',      'phone',       1, 0, 'var', '', 0 ],
        [ 'UserFax',        'Fax',        'fax',         1, 0, 'var', '', 0 ],
        [ 'UserMobile',     'Mobile',     'mobile',      1, 0, 'var', '', 0 ],
        [ 'UserRoom',       'Room',       'room',        1, 0, 'var', '', 0 ],
        [ 'UserStreet',     'Street',     'street',      1, 0, 'var', '', 0 ],
        [ 'UserZip',        'Zip',        'zip',         1, 0, 'var', '', 0 ],
        [ 'UserCity',       'City',       'city',        1, 0, 'var', '', 0 ],
        [ 'UserCountry',    'Country',    'country',     1, 0, 'var', '', 0 ],
        [ 'UserComment',    'Comment',    'comments',    1, 0, 'var', '', 0 ],
        [ 'ValidID',        'Valid',      'valid_id',    0, 1, 'int', '', 0 ],
    ],
    # default selections
    Selections => {
        UserTitle => {
            'Mr.' => 'Mr.',
            'Mrs.' => 'Mrs.',
        },
    },
};
```

4.3. Customizing the customer_user table in the OTRS DB

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:~#
```

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

1. 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'.

1.1. Neu

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

1.2. Öffnen

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

1.3. 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.

1.4. 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.

1.5. 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.

1.6. Zusammengefügt

Status von Tickets, die mit anderen Tickets zusammengefügt wurden.

1.7. Erfolgreich geschlossen

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

1.8. Erfolglos geschlossen

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

2. 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 KonfigurationsdateiKernel/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';
    [...]
```

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:~#
```

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 , especialy in the chapter on .

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" 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

You need a user ID to identify your key; the software constructs the user ID
from the Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

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

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

Passphrase: secret
Repeat passphrase: secret

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

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

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

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

linux:~$
```

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 wähen Sie danach dan 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-Konfigurationsverzeichnisses 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 OTRSmit 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:~$
```

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>
```

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>
```

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>
```

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>
```

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>
```

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)

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

2. 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". This is also affected by changes in the form (e.g. if the ticket is the queue "raw" and had a priority "3 normal", but then priority drop-down is selected and the priority is changed now to "5 very high" will also match).

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 allowing movement into a queue of only those tickets with ticket priority 5 stored in the database.

This example is very similar to the last one, but in this case only tickets in the queue "Raw" and with a priority "5 very high", both stored in the database will match. This kind of ACLs does not consider changes in the form before the ticket is really updated in the database.

```
# ticket acl
$Self->{TicketAcl}->{'100-Example-ACL'} = {
    # match properties
    PropertiesDatabase => {
        # current ticket match properties
        Ticket => {
            Queue => ['Raw'],
            Priority => ['5 very high'],
        }
    },
    # return possible options (white list)
    Possible => {
        # possible ticket options (white list)
        Ticket => {
            Queue => ['Alert'],
        },
    },
};
```

Please note that PropertiesDatabase is available since OTRS 3.1.5.

Beispiel 18.3. 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 18.4. ACL removing always state closed successful.

This example shows how it is possible to define negative filters (the state "closed successful" will be removed). You can also see that not defining match properties for a ticket will match any ticket, i. e. the ACL will always be applied. This may be useful if you want to hide certain values by default, and only enable them in special circumstances (e. g. if the agent is in a specific group).

```
$Self->{TicketAcl}->{'102-Third-ACL-Example'} = {
    # match properties
    Properties => {
        # current ticket match properties (match always)
    },
    # return possible options
    PossibleNot => {
        # possible ticket options
        Ticket => {
            State => ['closed successful'],
        },
    },
};
```

Beispiel 18.5. 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)'],
        },
    },
};
```

3. Reference

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

Beispiel 18.6. Reference showing all possible important ACL settings.

```
# ticket acl
$Self->{TicketAcl}->{'200-ACL-Reference'} = {
    # match properties (current values from the form)
    Properties => {

        # the used frontend module
        Frontend => {
            Action => ['AgentTicketPhone', 'AgentTicketEmail'],
        },
        # the logged in agent
        User => {
            UserLogin => ['some login'],
            # ...
            Group_rw => [
                'hotline',
            ],
            # ...
            Role => [
                'admin',
            ],
            # ...
        },
        # the logged in customer
        CustomerUser => {
            UserLogin => ['some login'],
            # ...
            Group_rw => [
                'hotline',
            ],
            # ...
            Role => [
                'admin',
            ],
            # ...
        },

        #
        # ticket properties
        #

        Queue => {
            Name     => ['Raw'],
            QueueID  => ['some id'],
            GroupID  => ['some id'],
            Email    => ['some email'],
            RealName => ['OTRS System'],
            # ...
        },
        Service => {
            ServiceID => ['some id'],
            Name      => ['some name'],
            ParentID  => ['some id'],
            # ...
        },
        Type => {
            ID   => ['some id'],
            Name => ['some name'],
            # ...
        },
```

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

    # match properties (existing values from the database)
```

```
    #   This section is available since OTRS 3.1.5.
    PropertiesDatabase => {
        # See section "Properties", the same config can be used here.
        # ...
    }

    # return possible options (white list)
    Possible => {
        # possible ticket options (white list)
        Ticket => {
            Queue => ['Hotline', 'Coordination'],
            State => ['some state'],
            Priority => ['5 very high'],
            DynamicField_Field1  => ['some value'],
            DynamicField_MyField => ['some value'],
            # ...
            NewOwner => ['some owner'],
            OldOwner => ['some owner'],
            # ...
        },

        # 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"
        # ...
    },
};
```

Anmerkung

While matching ACLs if CustomerUserID parameter sent, the ACL mechanism will compare the defined ACLs using the supplied CustomerUserID to gather the CustomerUser details to fill the CustomerUser hash and it also overrides the Customer information in the Ticket hash for the Properties match. On the other hand this calculations are also made for the PropertiesDatabase part, but using the Ticket Customer as the basis to gather the data.

Notice that in Customer Interface, the CustomerUserID is always sent with the current logged Customer User.

Be aware that in ticket search screens (AgentTicketSearch and CustomerTicketSearch) the only affected attributes by ACLs are the Dynamic Fields. This means that this screens you can not restrict any other attribute like ticket type, state, queue, etc.

Kapitel 19. Statistikmodul

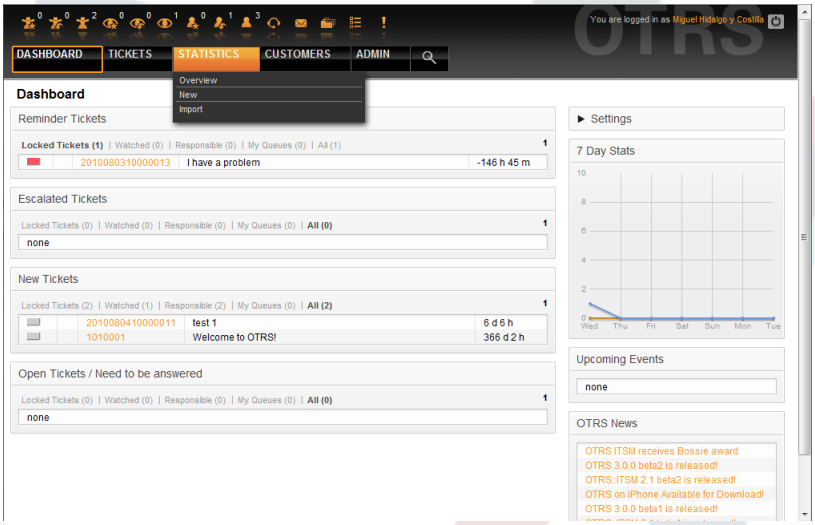
Das OTRS-Statistikmodul enthält Funktionen, um betriebliche Kennziffern zu überwachen und angepasste Reports der Verwendung von OTRS zu erstellen. In OTRS wird der Begriff "Stat/ Statistik" für einen Report benutzt, der verschiedene Indikatoren abbildet.

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).

1. Bedienung des Moduls durch den Agent

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

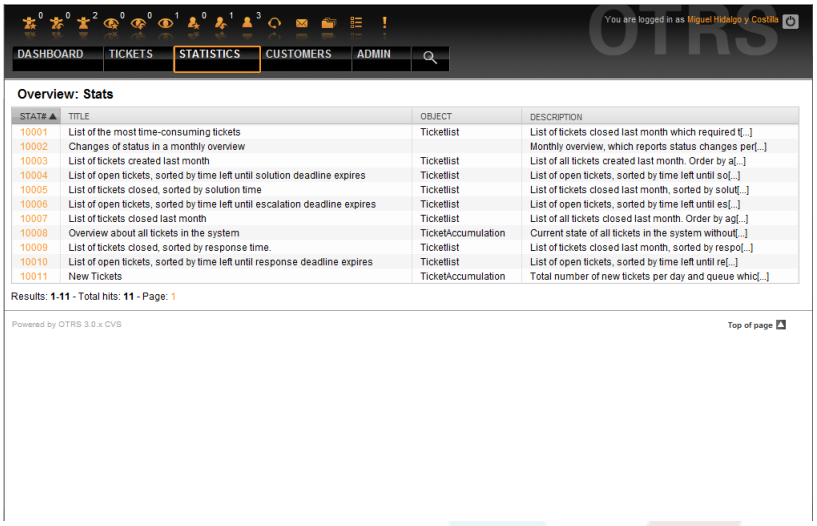


Folgende Optionen sind im Statistikmenü verfügbar:

- . Zeigt eine Liste vorkonfigurierter Reports an.
- . Benötigt "rw"-Rechte.
- . Benötigt "rw"-Rechte.

1.1. 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.



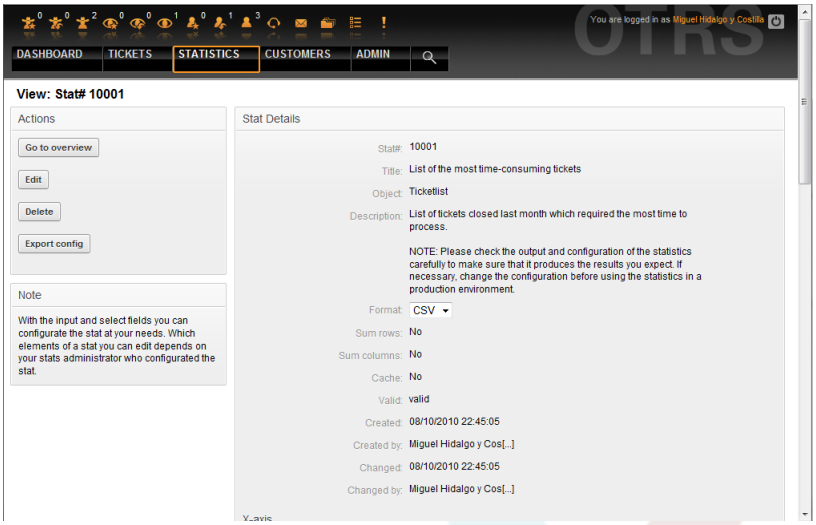
Folgende Informationen sind den in der Übersicht aufgelisteten Statistiken zu entnehmen:

- . Eindeutige Statistiknummer.
- . Titel der Statistik.
- . 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.
- . 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.

1.2. Ansehen und Generieren von Statistiken

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



Konfigurationseinstellungen für einen Report können in den Optionen der Ansichtsseite 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 Übersichtsliste der Reports.
  - . Die aktuelle Statistik bearbeiten (benötigt "rw"-Rechte).
  - . Die aktuelle Statistik löschen (benötigt "rw"-Rechte).
  - . 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:
  - . Die Nummer der Statistik.
  - . Titel der Statistik.
  - . Objekt, das für die Generierung der Statistik verwendet wird.
  - . Ein Auszug aus der Beschreibung der Statistik.
  - . Ausgabeformat der Statistik. Hierbei stehen je nach Konfiguration folgende Ausgabeformate zur Verfügung:
    - CSV.
    - Drucken.

  - Liniendiagramm.
  - Balkendiagramm.
  - Balkendiagramm (horizontal).
  - Punktdiagramm
  - Linienpunktdiagramm.
  - Flächendiagramm.
  - Tortendiagramm.

- ~~~~~~~~~~~~. 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.

- ~~~~~~~~~~~~~~~~~~. Gibt an, ob die Statistik durch eine Spalte ergänzt wird, deren Zellen die Summe der jeweiligen Zeile darstellt.

- ~~~~~~~~~~~~~~~~~~. Angabe, ob die Statistik durch eine Zeile ergänzt wird, deren Zellen die Summe der jeweiligen Spalte beinhaltet.

- ~~~~~. Gibt an, ob die errechneten Statistiken im Dateisystem gecacht werden.

- ~~~~~~. 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.

- ~~~~~~~~. Erstellungszeit der Statistik.

- ~~~~~~~~ ~~~. Agent, der die Statistik erstellt hat.

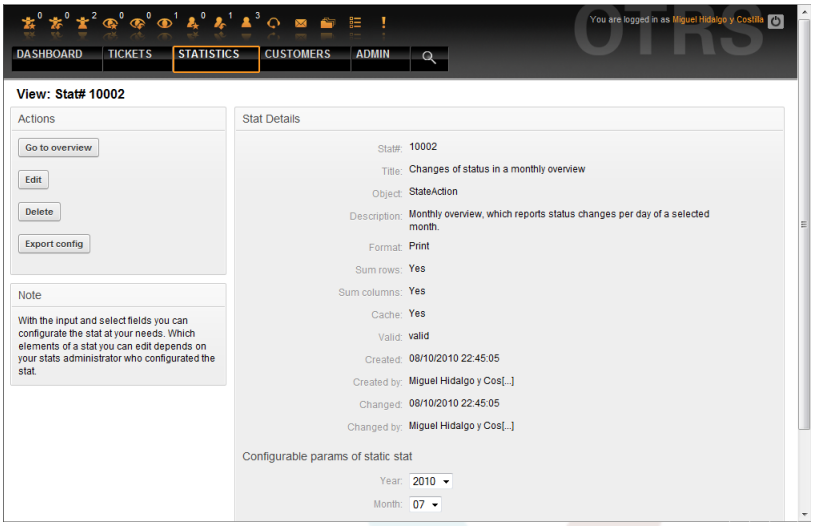
- ~~~~~~~~. Letzte Änderungszeit der Statistik.

- ~~~~~~~~ ~~~. Agent, der die Statistik zuletzt geändert hat.

- ~~~~~~~. 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:

  - ~~~~~~~ ~~~~~~~~~~ ~~~~~~~~~~. Statische Report-Generatoren können in die Statistik integriert werden (siehe unten).



- [illegible] Dynamische Statistiken können auf zweierlei Weise angezeigt werden:
  - [illegible]. In diesem Fall hat der Ersteller der Statistik dem Agent, der eine Statistik generieren will, keinen Freiraum für Veränderungen gelassen.
  - [illegible]. 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 Ansichtsseite wird nach Drücken des "Start"-Button wieder die Ansichtseite gezeigt und im Notificationbereich darauf hingewiesen, welche Eingaben falsch waren.

1.3. Bearbeiten / Neu

Agenten mit Schreibrechten können existierende Report-Konfigurationen bearbeiten, indem sie die Bearbeitungsmaske des Statistikmoduls aufrufen. Altertativ 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.

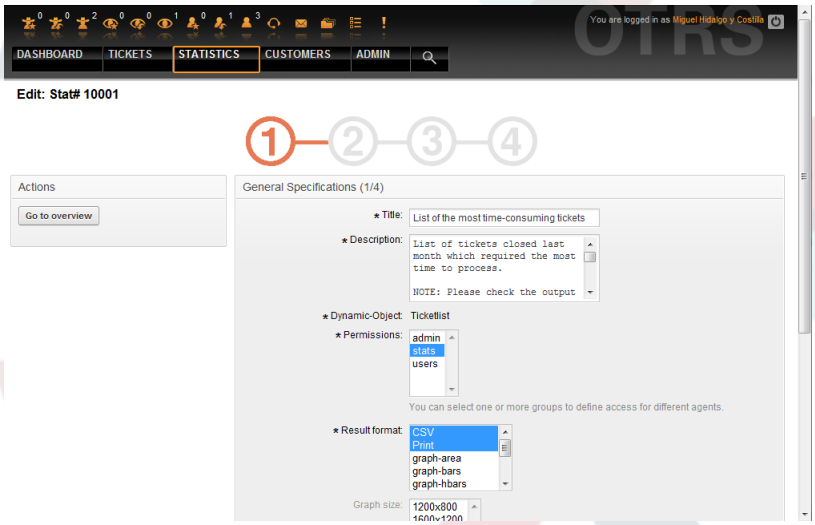
4. Festlegen der Einschränkungen des Reports.

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. ~~~~~~~~~~ ~~~~~~~. Dies ist die erste Seite des Assistenten.



~~~~~~~~~~
~~~~~~~~~~

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.

- ⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲. 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.

- ⚲⚲⚲⚲⚲⚲. Ausgabeformat der Statistik. Hierbei stehen je nach Konfiguration folgende Ausgabeformate zur Verfügung:

  - CSV.
  - Drucken.
  - Liniendiagramm.
  - Balkendiagramm.
  - Balkendiagramm (horizontal).
  - Punktdiagramm.
  - Linienpunktdiagramm.
  - Flächendiagramm.
  - Tortendiagramm.

- ⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲. 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.

- ⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲. Wird im "Ja" ausgewählt, wird die Statistik durch eine Spalte ergänzt, deren Zellen die Summe der jeweiligen Zeile darstellt.

- ⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲⚲. Angabe, ob die Statistik durch eine Zeile ergänzt wird, deren Zellen die Summe der jeweiligen Spalte beinhaltet.

- ⚲⚲⚲⚲⚲. 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.

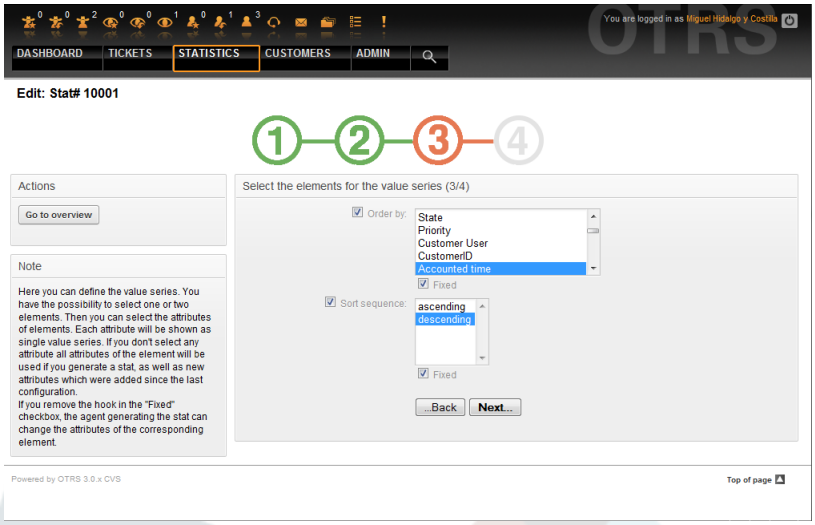
Sollte eine Statistik zwischenzeitlich wieder bearbeitet werden, werden alle gecachten Daten wieder gelöscht.

- ~~~~~~. 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. ~~~~~~

3. ⁓⁓⁓⁓⁓⁓⁓⁓⁓⁓⁓.

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).



⁓⁓⁓⁓⁓ ⁓⁓⁓⁓⁓⁓⁓⁓⁓⁓ ⁓⁓⁓ ⁓⁓⁓⁓⁓⁓⁓⁓⁓⁓⁓⁓⁓

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

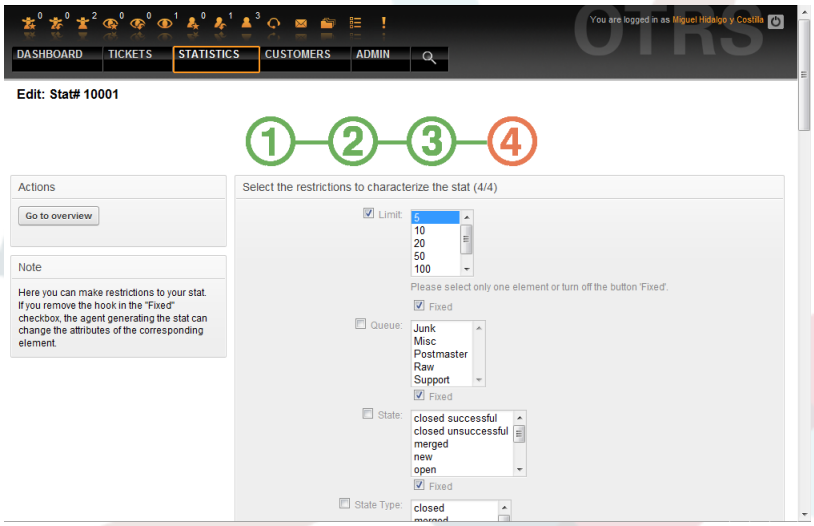
Die Auswahl von drei oder mehr Elementen ist nicht sinnvoll und wird deshalb durch eine Fehlermeldung verhindert.

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.

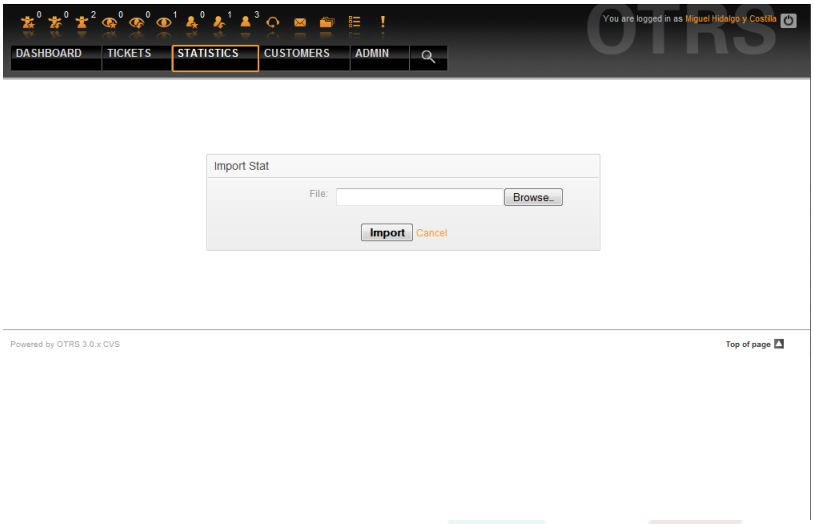
4. . Vierter und letzter Schritt bei der Vorkonfiguration ist das Festlegen der Einschränkungen. Hier ist es möglich die Ergebnisse einer Statistik durch die Angabe von Bedingungen einzuschränken. Vergleichbar ist dies mit Eingaben in eine Such-Oberfläche.



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

1.4. 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).



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.

2. 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.

2.1. 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:

- . Darf Statistiken konfigurieren.
- . 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.

2.2. 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.

3. 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`).

3.1. 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.

3.2. 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`

3.3. Caching

Bei der Vorkonfiguration von Statistiken kann eingestellt werden, ob die Ergebnisse einer Statistik gecachet 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 gecacheten 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.

3.4. 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
```

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
```

3.5. 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.

3.6. 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/`.

3.7. 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;
```

3.8. 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 acts as a server listening to requests from the External System, processing the information, performing the requested action, and answering the request.
- ~~~~~ ~~ ~~~~~~~~~~ OTRS acts as a client collecting information, sending the request to the Remote System, and waiting for the response.

1. Generic Interface Layers

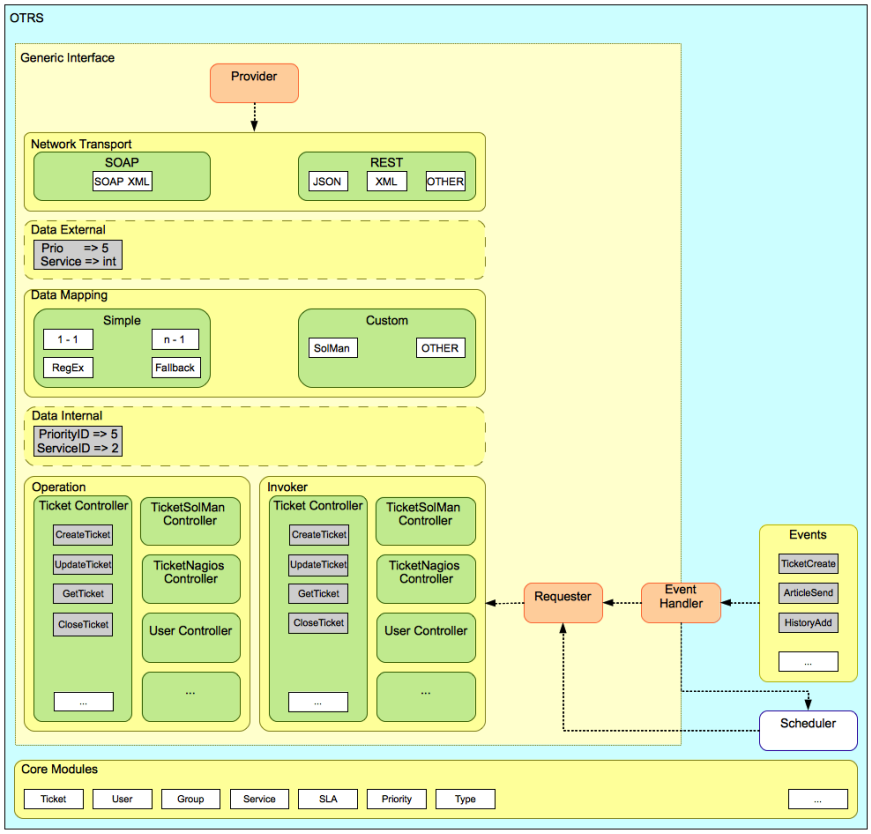
Generic Interface is build based on a layer model, to be flexible and easy to customize.

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.



1.1. 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 catched by the event handler and depending on the configuration the event will be processed directly by the requester object or delegated to the Scheduler (a separated deamon designed to process tasks asynchronously).

1.2. 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.

"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.

1.3. 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.

1.4. 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.

1.5. 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.

2. 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:

2.1. OTRS as Provider

2.1.1. 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 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 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.

2.1.2. OTRS Response:

1. Operation
   - Returns result data to the provider.
2. ~~~~ ~~~~~~~~~~
   - 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 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).

2.2. OTRS as Requester

2.2.1. 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 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 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

2.2.2. Remote Response:

1. Network transport
   - Receives the response and decodes the data payload.

- Returns the data to the requester.

2. ~~~~ ~~~~~~~~~

   - Data as received from the Remote System

3. Mapping

   - The data is transformed form 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 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.

3. 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.

4. 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.

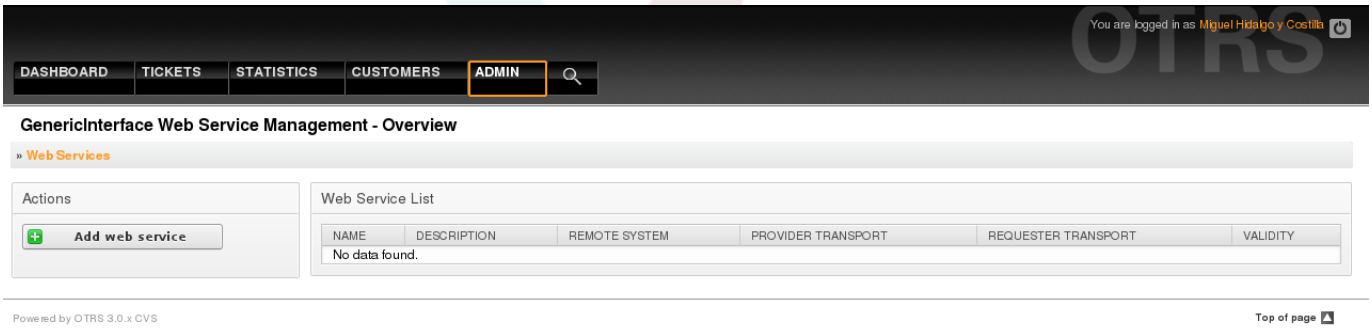
4.1. Web Service Overview

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.



4.2. 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

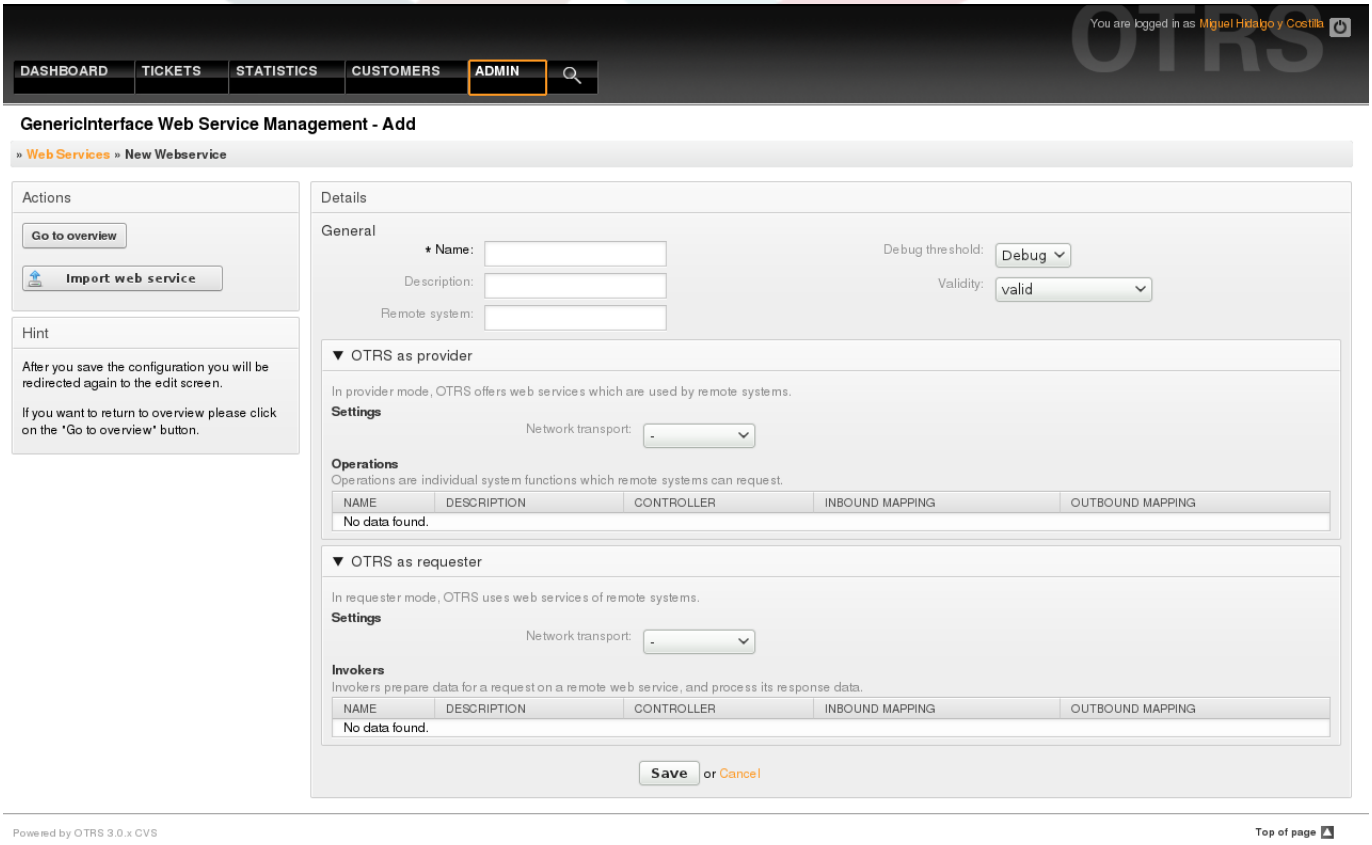
It is also possible to define the network transport protocol for "OTRS as Provider" and "OTRS as requester".

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.



4.3. 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.

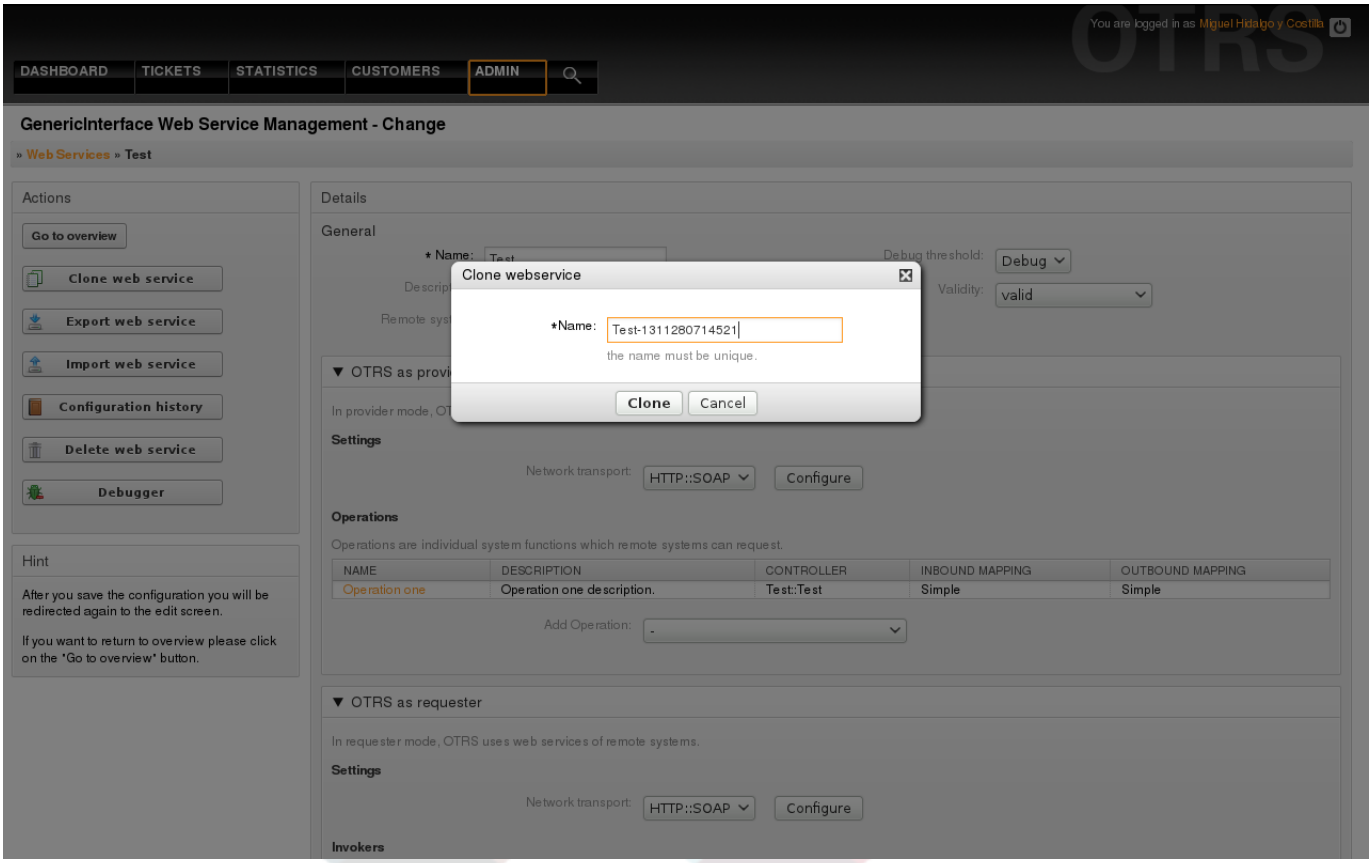
> **Anmerkung**
>
> "Configuration history" and "Debugger" will lead you to different screens.

4.3.1. 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**
>
> 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.



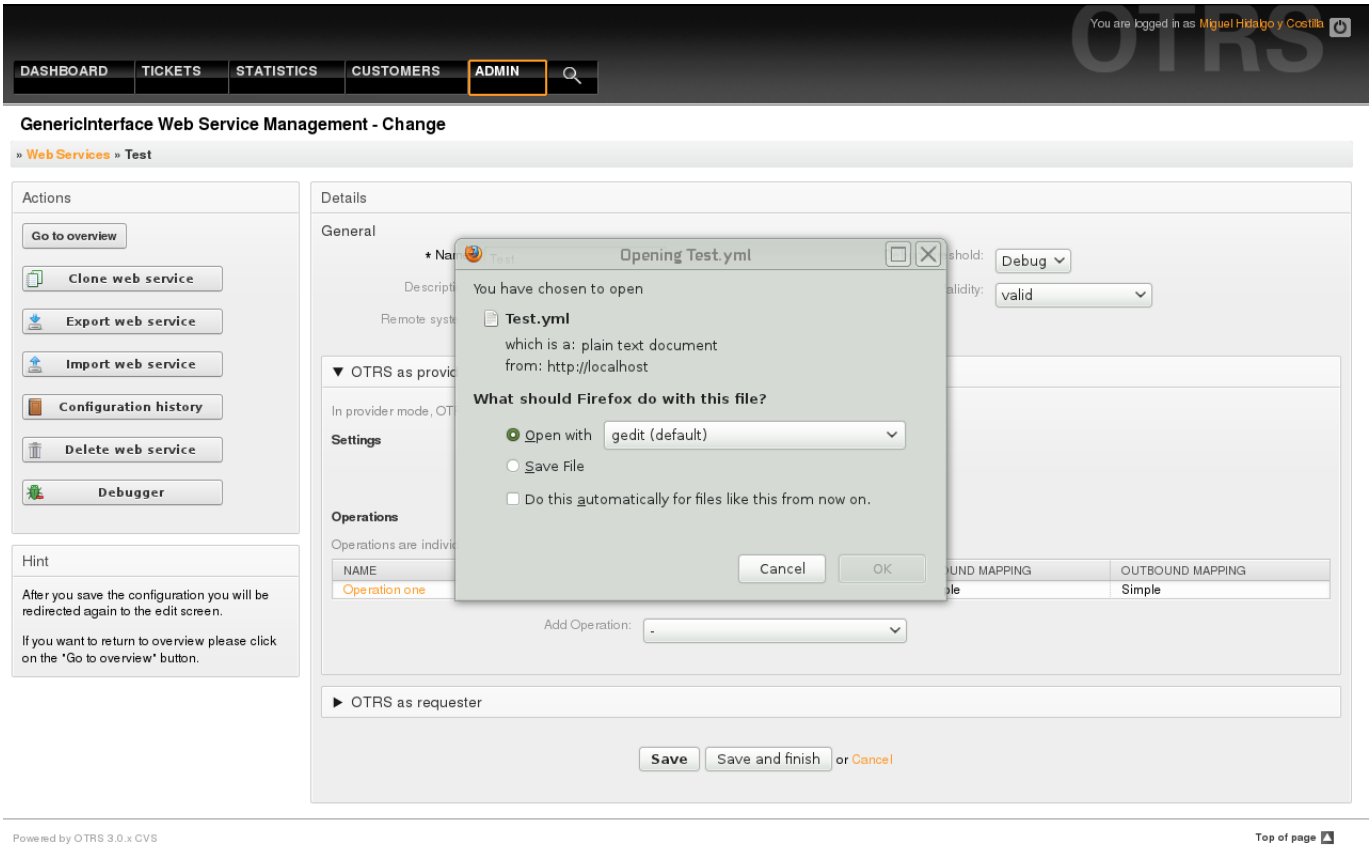
4.3.2. Web Service Export

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.

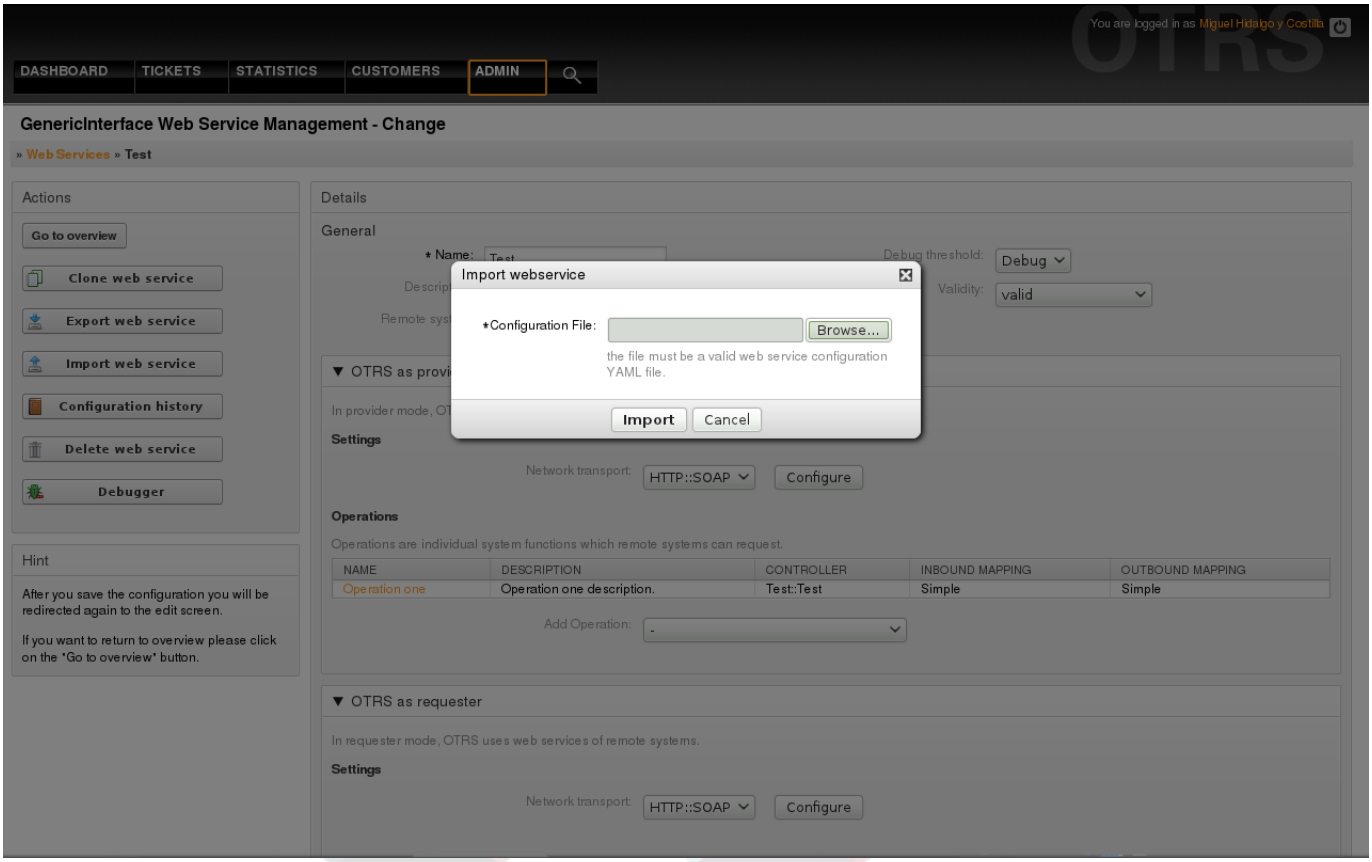


4.3.3. Web Service Import

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.



4.3.4. 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.

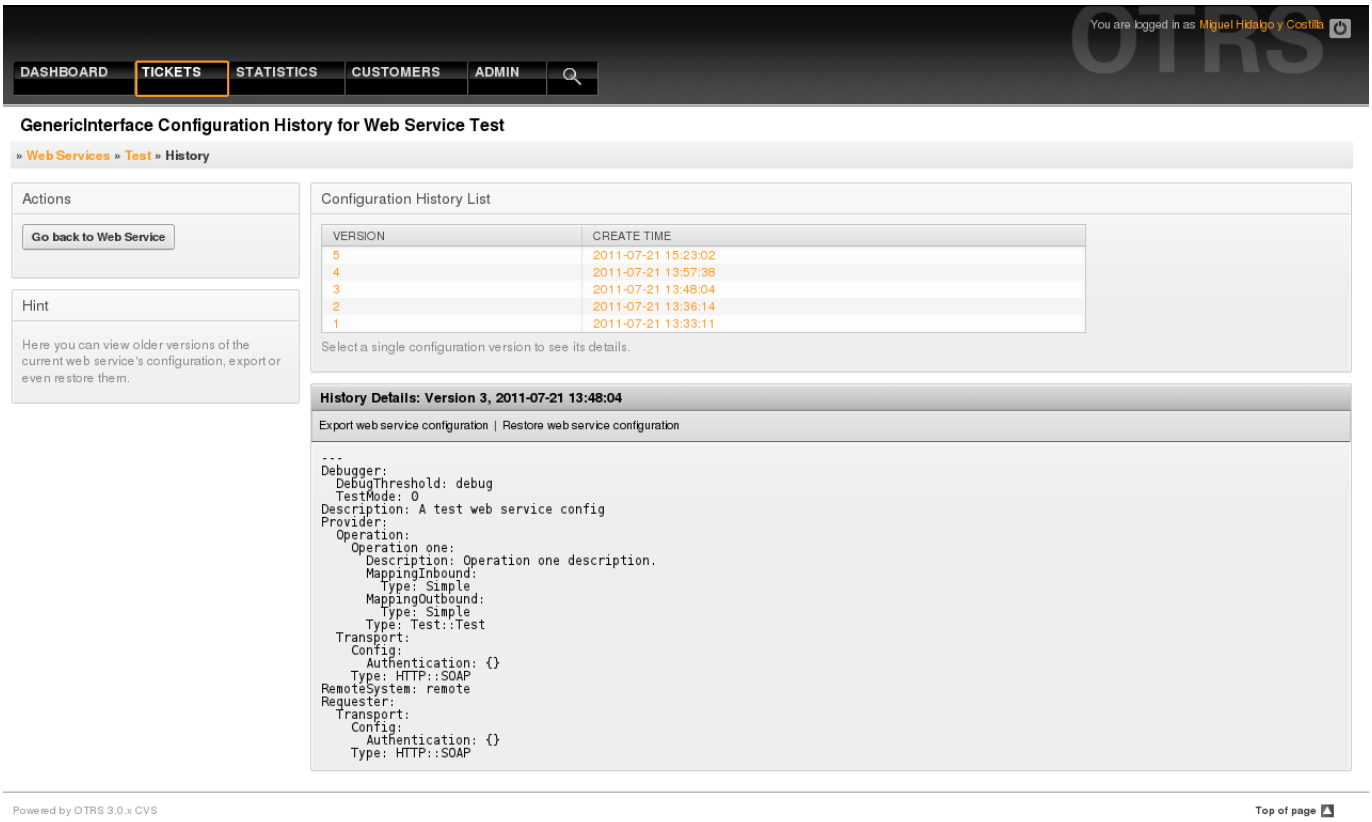
The "Export web service configuration" behaves exactly as the "Export web service" feature in the web service change screen. For more information refer to that section.

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.



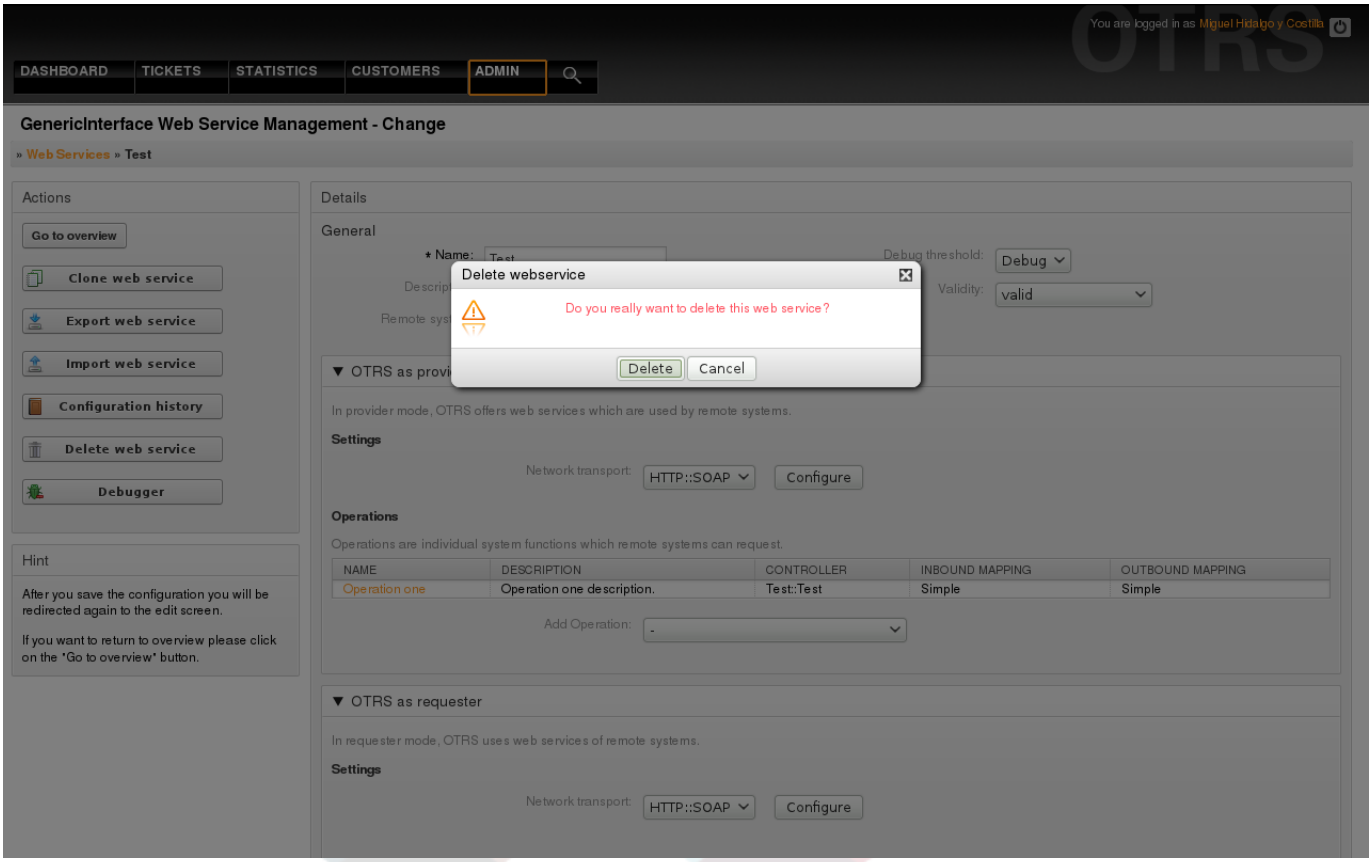
4.3.5. Web Service Delete

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.

Clink 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.



4.3.6. 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.

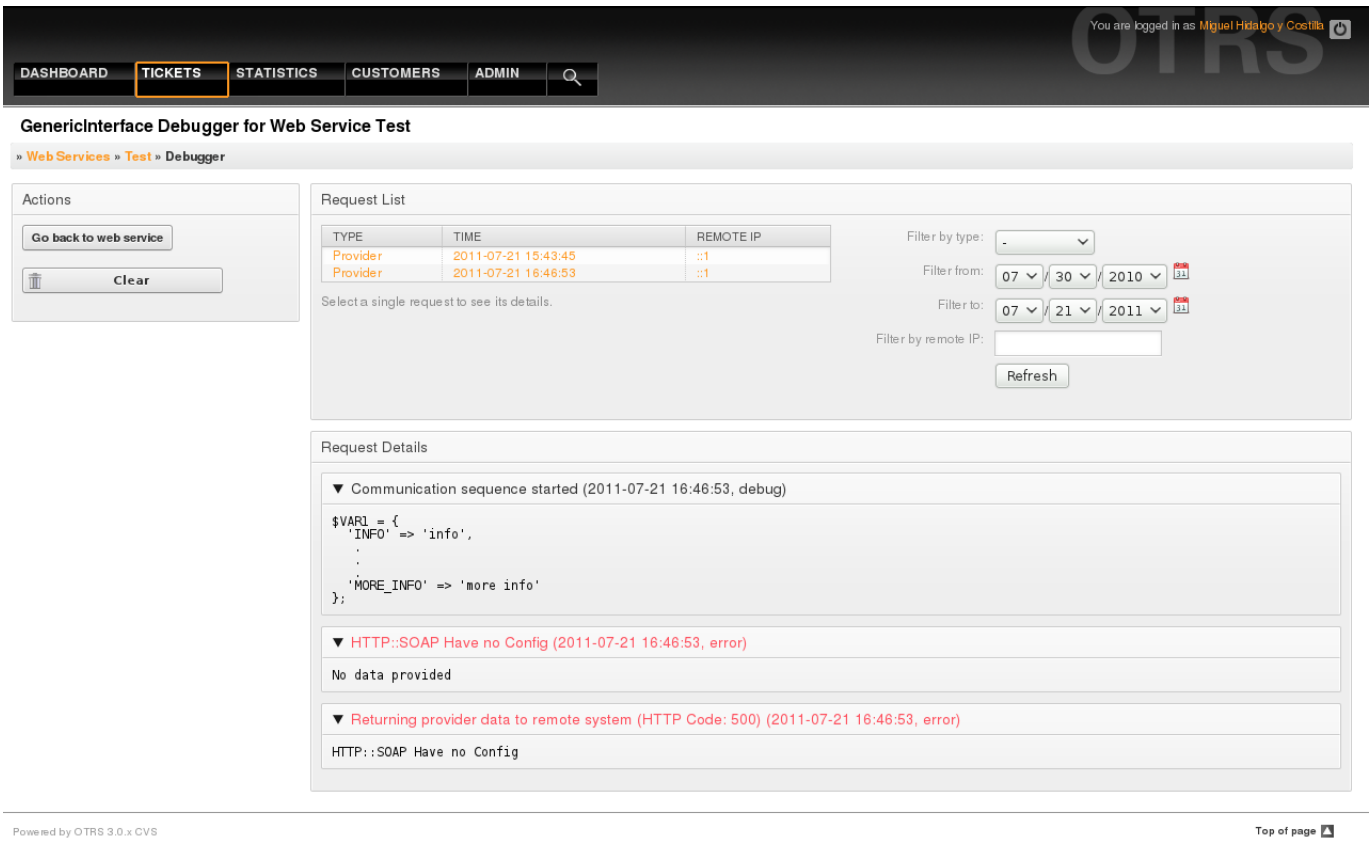
After filter settings are set, push the "Refresh" button and a new list will be displayed meeting your search criteria.

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.



4.3.7. 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".

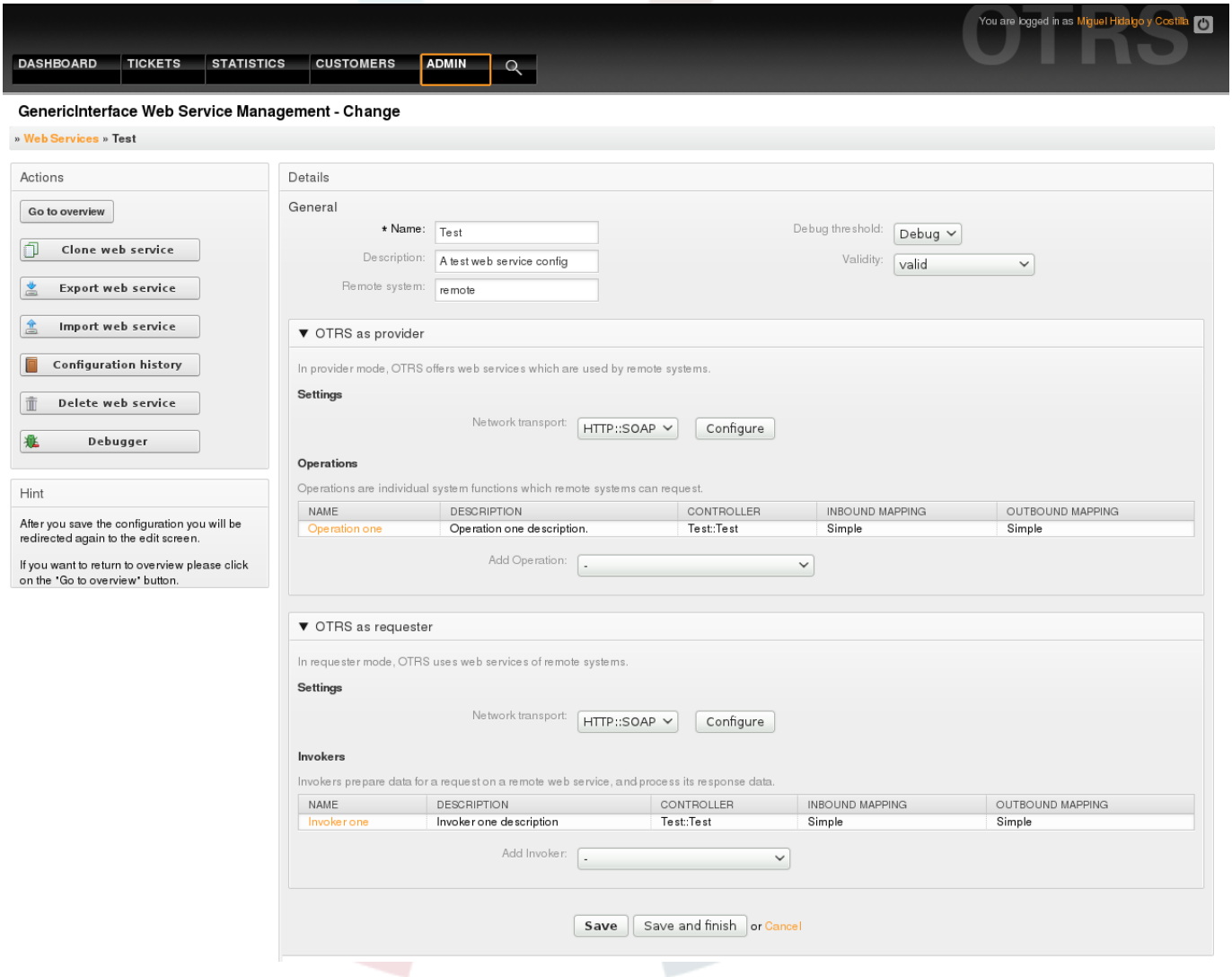
The web service configuration needs to be saved on each level. This means that if a setting is changed, links to other, deeper parts of the configuration will be disabled forcing to save the current configuration level. After saving the disabled links will be re-enabled again allowing you to continue 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.



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.

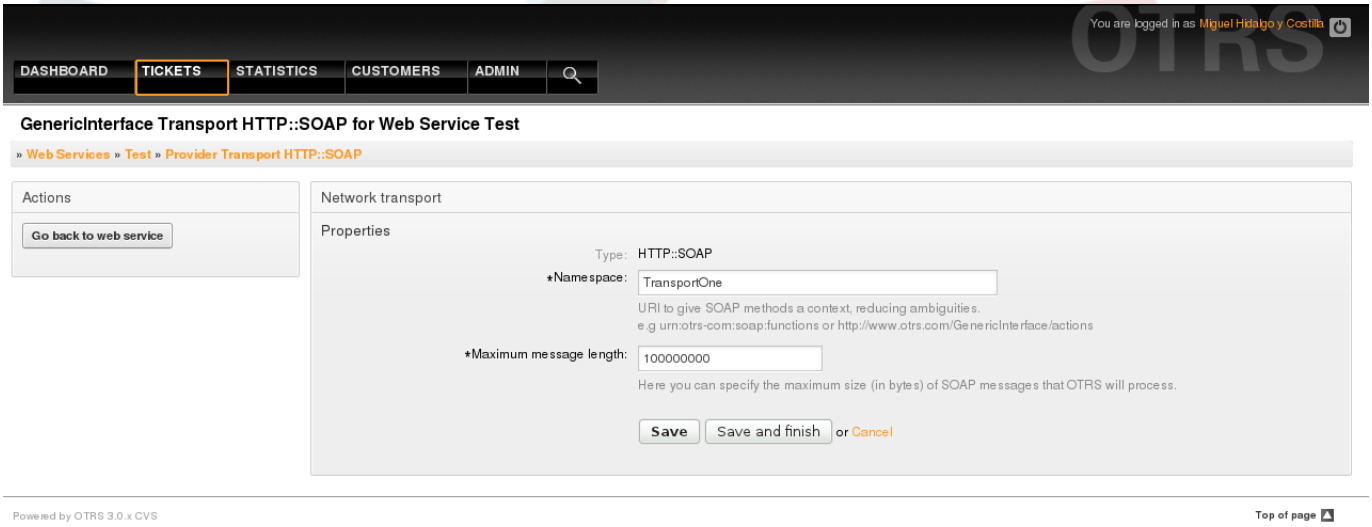
"Save and Finish" will store the current configuration level in the database and it will return to the previous screen in the configuration hierarchy (to the immediate upper configuration level).

"Cancel" will discard any configuration change to the current configuration level and will return to the previous screen in the configuration hierarchy.

4.3.7.1. 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.



4.3.7.2. 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

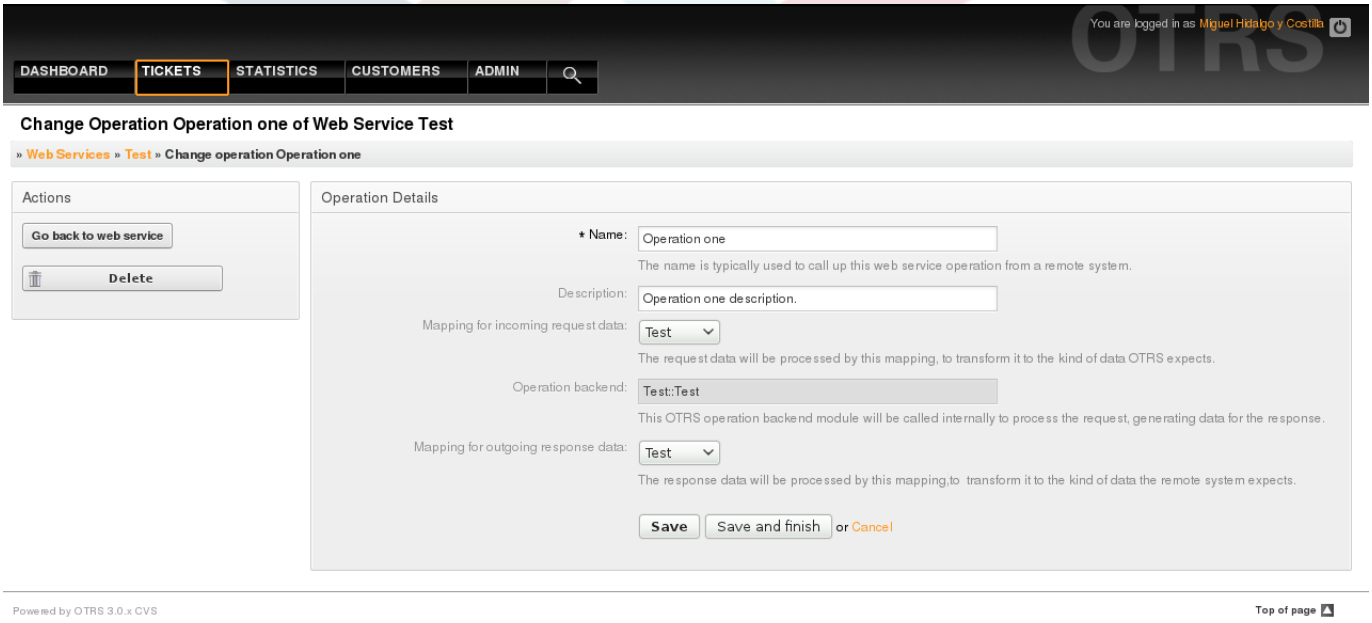
or invokers, normally operations from the same controller need similar settings and shares the same configuration dialog. But each operation can have independent configuration dialogs if needed.

Name, Description, Backend, and Mappings are fields that normally appear on every operation, other special fields can appear in non default configuration dialogs to fulfill specific needs of the operation.

Normally there are two mapping configuration sections on each operation, one for the incoming data and another one for the outgoing data. You can choose different mapping types (backends) for each mapping direction, since their configuration is independent from each other and also independent from the operation backend. The normal and most common practice is that the operation uses 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 it configuration or "Cancel" to close the delete dialog.



4.3.7.3. 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:

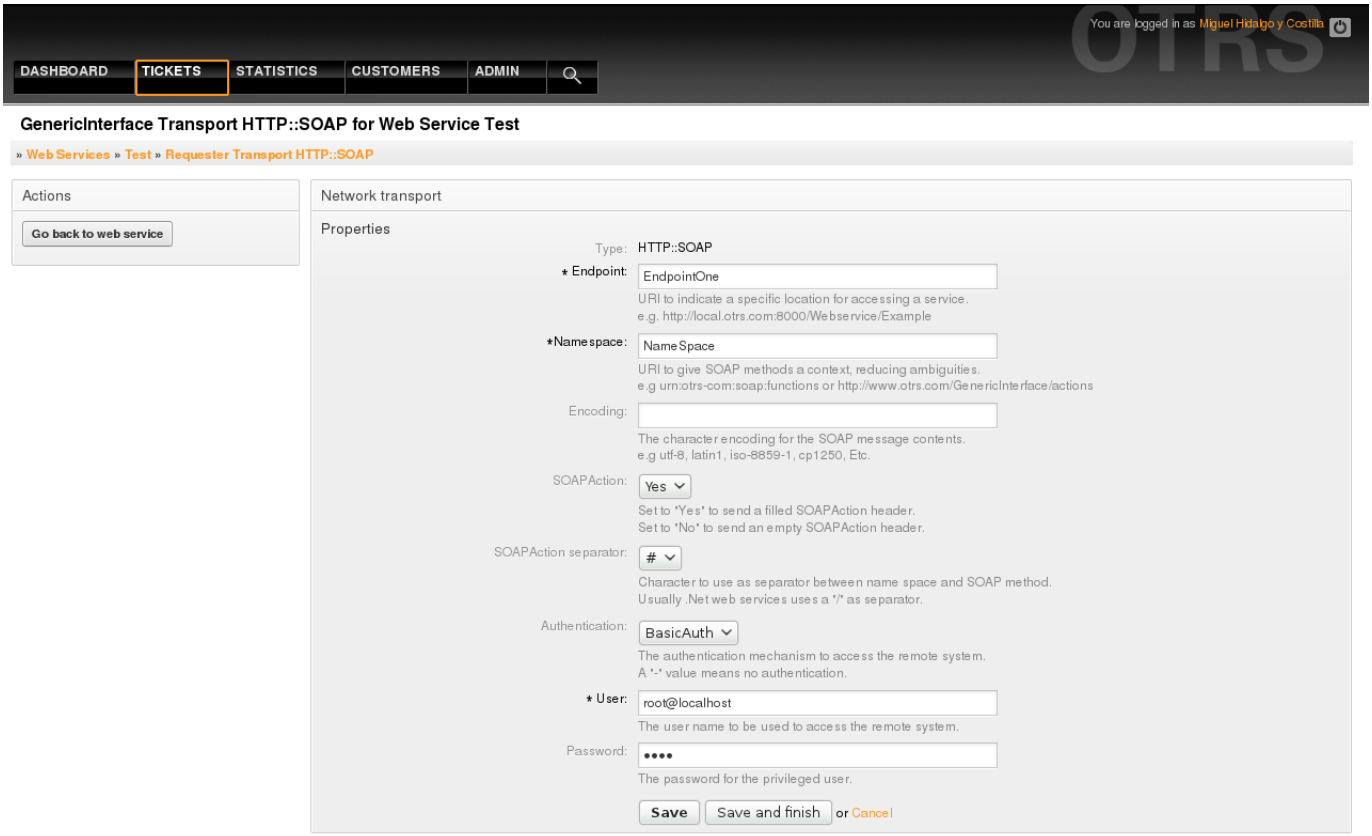
- Encoding (such as utf-8, latin1, iso-8859-1, cp1250, etc) for the SOAP message.
- SOAPAction Header: you can use this to send an empty or filled SOAPAction header. Set to "No" and the SOAPAction header on the SOAP message will be an empty string, or set to "Yes" to send the soap action in Namespace#Action 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.



4.3.7.4. 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 to fulfill special needs of the invoker.

Normally there are two mapping configuration sections for each invoker, one for the incoming data and another one for the outgoing data. You can choose different mapping types (backends) for each mapping direction, since their configuration is independent from each other and also independent from the invoker backend. The normal and most common practice is that the invoker uses the same mapping type in both cases, with inverted configuration. The complete mapping configuration is done in a separate screen, which depends on the mapping type.

The invoker backend is pre-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.

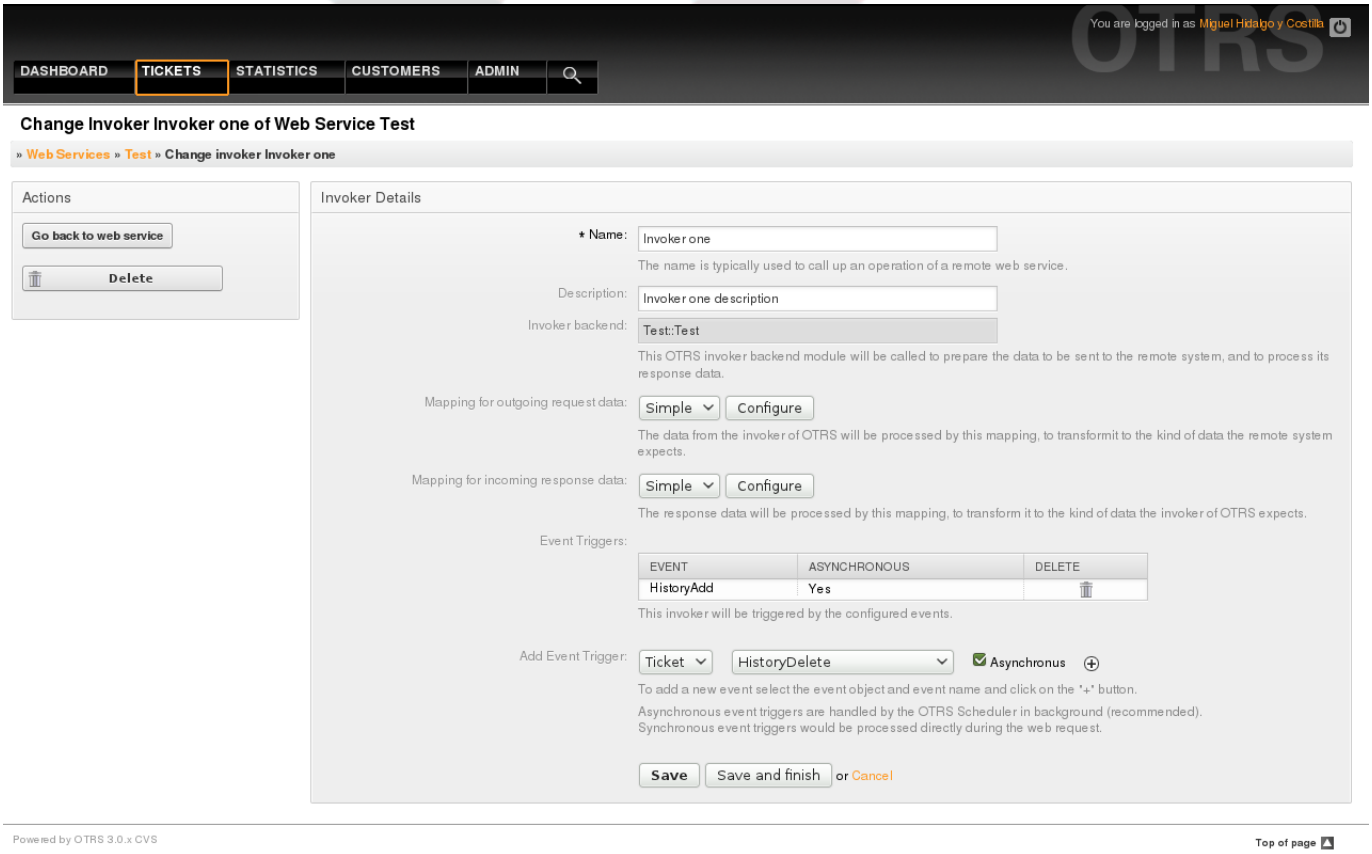
Anmerkung

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.



4.3.7.5. Web Service Mapping

There are cases where you need to transform the data from one format to another (map or change data structure), because normally a web service is used to interact with a Remote System, that is highly probable that is not another OTRS system and / or could not understand the OTRS data structures and values. In these cases some or all values 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.

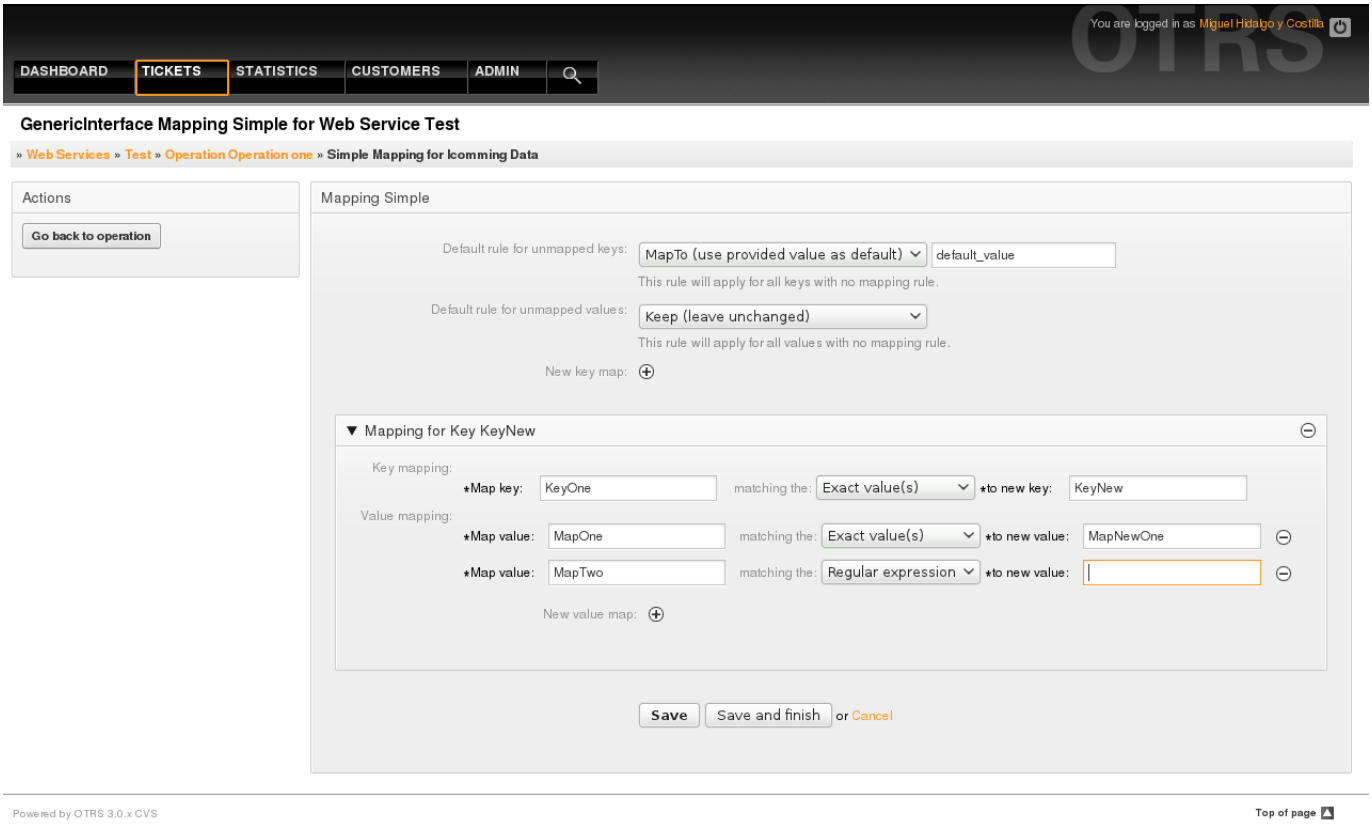
Clicking on the "+" button for new key map, will display a new box for a single mapping configuration. You can add as many key mappings as needed. Just click on the "+" button again and a new mapping box will appear below the existing one. From this mapping boxes you can define a map for a single key, 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.



5. 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.

5.1. 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.

5.2. 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 ~~~~~~~~ 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.

6. 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 test operation
      MappingInbound: {}
      MappingOutbound: {}
      RemoteSystemGuid: ''
      Type: Test::Test
    Test:
      Description: This is a test operation
      MappingInbound:
        Config:
          KeyMapDefault:
            MapTo: ''
            MapType: Keep
          KeyMapExact:
            Prio: Priority
          ValueMap:
            Priority:
              ValueMapExact:
                Critical: 5 Very High
                Information: 1 Very Low
                Warning: 3 Normal
          ValueMapDefault:
            MapTo: 3 Normal
            MapType: MapTo
```

```
        Type: Simple
      MappingOutbound:
        Config:
          KeyMapDefault:
            MapTo: ''
            MapType: Ignore
          KeyMapExact:
            Priority: Prio
          ValueMap:
            Prio:
              ValueMapExact:
                1 Very Low: Information
                3 Normal: Warning
                5 Very High: Critical
          ValueMapDefault:
            MapTo: ''
            MapType: Ignore
        Type: Simple
      Type: Test::Test
  Transport:
    Config:
      MaxLength: 10000000
      NameSpace: http://www.example.com/actions
    Type: HTTP::SOAP
RemoteSystem: remote.system.description.example.com
Requester:
  Invoker:
    Test:
      Description: This is a test invoker
      Events:
        - Asynchronous: 1
          Event: TicketCreate
        - Asynchronous: 0
          Event: ArticleUpdate
      MappingInbound:
        Type: Simple
      MappingOutbound:
        Type: Simple
      Type: Test::Test
  Transport:
    Config:
      Authentication:
        Password: '*******'
        Type: BasicAuth
        User: otrs
      Encoding: utf-8
      Endpoint: http://www.example.com:8080/endpoint
      NameSpace: http://www.example.com/actions
      SOAPAction: Yes
      SOAPActionSeparator: '#'
    Type: HTTP::SOAP
```

6.1. Configuration Details

6.1.1. 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.

6.1.2. 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.

6.1.3. Provider

- Operation: a container for each operation settings.
- Transport: a container for provider network transport settings.

6.1.3.1. Operation

- <OperationName>: Unique name for the operation, container for its own operation settings (cardinality 0..n, but not duplicate).

6.1.3.1.1. <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.

6.1.3.1.1.1. 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.

6.1.3.1.1.1.1. 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).

6.1.3.1.1.1.1.1. 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.

6.1.3.1.1.1.1.2. ValueMapDefault

Similar to KeyMapDefault.

6.1.3.1.1.1.1.3. KeyMapExact

- <oldkey>: <newkey> (cardinality 0 .. n but not duplicate).

6.1.3.1.1.1.1.4. KeyMapRegEx

- <oldkey(RegEx)>: <newkey> (cardinality 0 .. n but no duplicates).

6.1.3.1.1.1.1.5. ValueMap

- <newkey>: a container for value mappings for this new key (cardinality depends on the new keys from KeyMapExact and KeyMapRegEx).

6.1.3.1.1.1.1.5.1. <newkey>

- ValueMapExact: a container for all exact value mappings (cardinality 0 .. 1).
- ValueMapRegEx: a container for all regular expression value mappings (cardinality 0 .. 1).

6.1.3.1.1.1.1.5.1.1. valueMapExact

- <oldvalue>: <newvalue> (cardinality 0 .. n but not duplicate).

6.1.3.1.1.1.1.5.1.2. ValueMapRegEx

- <oldvalue(RegEx)>: <newvalue> (cardinality 0 .. n but not duplicate).

6.1.3.1.1.2. MappingOutbound

Same as MappingInbound.

6.1.3.1.1.3. 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.

6.1.3.1.1.3.1. 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.

6.1.4. Requester

- Invoker: a container for each invokers' settings.

- Transport: a container for requester network transport settings.

6.1.4.1. Invoker

- <InvokerName>: Unique name for the invoker, container for its own invoker settings (cardinality 0..n, but not duplicate).

6.1.4.1.1. <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.

6.1.4.1.1.1. Events

- ⁓⁓⁓⁓ ⁓⁓⁓⁓⁓⁓⁓⁓: (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

6.1.4.1.1.2. MappingInbound

Same as Operation MappingInbound

6.1.4.1.1.3. MappingOutbound

Same as Operation MappingInbound.

6.1.4.1.1.4. 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.

6.1.4.1.1.4.1. 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.

6.1.4.1.1.4.1.1. 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.

7. 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.

7.1. Bundled Connectors

Included with this version of OTRS the following connectors are ready to be used.

- Session
- Ticket

7.1.1. Session Connector

This connector is capable to create a valid SessionID that can be used in any other operation.

Provides:

- Operations:
  - SessionCreate

7.1.1.1. Operations

7.1.1.1.1. 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>
```

7.1.2. Ticket Connector

This connector supplies the basic functionality to interact with tickets

Provides:

- Operations:
  - TicketCreate
  - TicketUpdate
  - TicketGet
  - TicketSearch

7.1.2.1. Operations

7.1.2.1.1. 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>
```

7.1.2.1.2. 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>
```

7.1.2.1.3. 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:-->
         <ArticleSenderType>?</ArticleSenderType>
         <!--Optional:-->
         <ArticleOrder>?</ArticleOrder>
         <!--Optional:-->
         <ArticleLimit>?</ArticleLimit>
         <!--Optional:-->
         <Attachments>?</Attachments>
      </TicketGet>
```

7.1.2.1.4. 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:-->
         <TypeIDs>?</TypeIDs>
         <!--Zero or more repetitions:-->
         <States>?</States>
         <!--Zero or more repetitions:-->
         <StateIDs>?</StateIDs>
         <!--Zero or more repetitions:-->
         <StateType>?</StateType>
         <!--Zero or more repetitions:-->
         <StateTypeIDs>?</StateTypeIDs>
         <!--Zero or more repetitions:-->
         <Priorities>?</Priorities>
         <!--Zero or more repetitions:-->
         <PriorityIDs>?</PriorityIDs>
         <!--Zero or more repetitions:-->
         <Services>?</Services>
         <!--Zero or more repetitions:-->
         <ServiceIDs>?</ServiceIDs>
         <!--Zero or more repetitions:-->
         <SLAs>?</SLAs>
         <!--Zero or more repetitions:-->
         <SLAIDs>?</SLAIDs>
         <!--Zero or more repetitions:-->
         <Locks>?</Locks>
         <!--Zero or more repetitions:-->
         <LockIDs>?</LockIDs>
         <!--Zero or more repetitions:-->
         <OwnerIDs>?</OwnerIDs>
         <!--Zero or more repetitions:-->
         <ResponsibleIDs>?</ResponsibleIDs>
         <!--Zero or more repetitions:-->
         <WatchUserIDs>?</WatchUserIDs>
         <!--Zero or more repetitions:-->
         <CustomerID>?</CustomerID>
         <!--Zero or more repetitions:-->
         <CustomerUserLogin>?</CustomerUserLogin>
         <!--Zero or more repetitions:-->
         <CreatedUserIDs>?</CreatedUserIDs>
         <!--Zero or more repetitions:-->
         <CreatedTypes>?</CreatedTypes>
         <!--Zero or more repetitions:-->
         <CreatedTypeIDs>?</CreatedTypeIDs>
         <!--Zero or more repetitions:-->
         <CreatedPriorities>?</CreatedPriorities>
         <!--Zero or more repetitions:-->
         <CreatedPriorityIDs>?</CreatedPriorityIDs>
         <!--Zero or more repetitions:-->
         <CreatedStates>?</CreatedStates>
         <!--Zero or more repetitions:-->
```

```
         <CreatedStateIDs>?</CreatedStateIDs>
         <!--Zero or more repetitions:-->
         <CreatedQueues>?</CreatedQueues>
         <!--Zero or more repetitions:-->
         <CreatedQueueIDs>?</CreatedQueueIDs>
         <!--Zero or more repetitions:-->
         <DynamicFields>
            <!--You have a MANDATORY CHOICE of the next 6 items at this level-->
            <!--Optional:-->
            <Equals>?</Equals>
            <!--Optional:-->
            <Like>?</Like>
            <!--Optional:-->
            <GreaterThan>?</GreaterThan>
            <!--Optional:-->
            <GreaterThanEquals>?</GreaterThanEquals>
            <!--Optional:-->
            <SmallerThan>?</SmallerThan>
            <!--Optional:-->
            <SmallerThanEquals>?</SmallerThanEquals>
         </DynamicFields>
         <!--Optional:-->
         <Ticketflag>
            <!--Optional:-->
            <Seen>?</Seen>
         </Ticketflag>
         <!--Optional:-->
         <From>?</From>
         <!--Optional:-->
         <To>?</To>
         <!--Optional:-->
         <Cc>?</Cc>
         <!--Optional:-->
         <Subject>?</Subject>
         <!--Optional:-->
         <Body>?</Body>
         <!--Optional:-->
         <FullTextIndex>?</FullTextIndex>
         <!--Optional:-->
         <ContentSearch>?</ContentSearch>
         <!--Optional:-->
         <ConditionInline>?</ConditionInline>
         <!--Optional:-->
         <ArticleCreateTimeOlderMinutes>?</ArticleCreateTimeOlderMinutes>
         <!--Optional:-->
         <ArticleCreateTimeNewerMinutes>?</ArticleCreateTimeNewerMinutes>
         <!--Optional:-->
         <ArticleCreateTimeNewerDate>?</ArticleCreateTimeNewerDate>
         <!--Optional:-->
         <ArticleCreateTimeOlderDate>?</ArticleCreateTimeOlderDate>
         <!--Optional:-->
         <TicketCreateTimeOlderMinutes>?</TicketCreateTimeOlderMinutes>
         <!--Optional:-->
         <ATicketCreateTimeNewerMinutes>?</ATicketCreateTimeNewerMinutes>
         <!--Optional:-->
         <TicketCreateTimeNewerDate>?</TicketCreateTimeNewerDate>
         <!--Optional:-->
         <TicketCreateTimeOlderDate>?</TicketCreateTimeOlderDate>
         <!--Optional:-->
         <TicketChangeTimeOlderMinutes>?</TicketChangeTimeOlderMinutes>
         <!--Optional:-->
         <TicketChangeTimeNewerMinutes>?</TicketChangeTimeNewerMinutes>
         <!--Optional:-->
         <TicketChangeTimeNewerDate>?</TicketChangeTimeNewerDate>
         <!--Optional:-->
         <TicketChangeTimeOlderDate>?</TicketChangeTimeOlderDate>
         <!--Optional:-->
```

```
        <TicketCloseTimeOlderMinutes>?</TicketCloseTimeOlderMinutes>
        <!--Optional:-->
        <TicketCloseTimeNewerMinutes>?</TicketCloseTimeNewerMinutes>
        <!--Optional:-->
        <TicketCloseTimeNewerDate>?</TicketCloseTimeNewerDate>
        <!--Optional:-->
        <TicketCloseTimeOlderDate>?</TicketCloseTimeOlderDate>
        <!--Optional:-->
        <TicketPendingTimeOlderMinutes>?</TicketPendingTimeOlderMinutes>
        <!--Optional:-->
        <TicketPendingTimeNewerMinutes>?</TicketPendingTimeNewerMinutes>
        <!--Optional:-->
        <TicketPendingTimeNewerDate>?</TicketPendingTimeNewerDate>
        <!--Optional:-->
        <TicketPendingTimeOlderDate>?</TicketPendingTimeOlderDate>
        <!--Optional:-->
        <TicketEscalationTimeOlderMinutes>?</TicketEscalationTimeOlderMinutes>
        <!--Optional:-->
        <TTicketEscalationTimeNewerMinutes>?</TTicketEscalationTimeNewerMinutes>
        <!--Optional:-->
        <TicketEscalationTimeNewerDate>?</TicketEscalationTimeNewerDate>
        <!--Optional:-->
        <TicketEscalationTimeOlderDate>?</TicketEscalationTimeOlderDate>
        <!--Optional:-->
        <ArchiveFlags>?</ArchiveFlags>
        <!--Zero or more repetitions:-->
        <OrderBy>?</OrderBy>
        <!--Zero or more repetitions:-->
        <SortBy>?</SortBy>
        <!--Zero or more repetitions:-->
        <CustomerUserID>?</CustomerUserID>
      </TicketSearch>
```

7.2. Examples:

7.2.1. 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: ''
```

7.2.2. 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 scipt.

```
#!/usr/bin/perl -w
# --
# otrs.SOAPRequest.pl - sample to send a SOAP request to OTRS Generic Interface Ticket
 Connector
# Copyright (C) 2001-2013 OTRS AG, http://otrs.org/
# --
# $Id: genericinterface-connectors.xml,v 1.8.2.1 2013-01-10 15:33:57 ub 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 know as ~~~~~~~~ in Unix / Linux systems or as ~~~~~~~~ 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).

1. Scheduler Graphical Interface

The Scheduler is not visible in the OTRS Graphical User Interface unless it stops running.

1.1. 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.

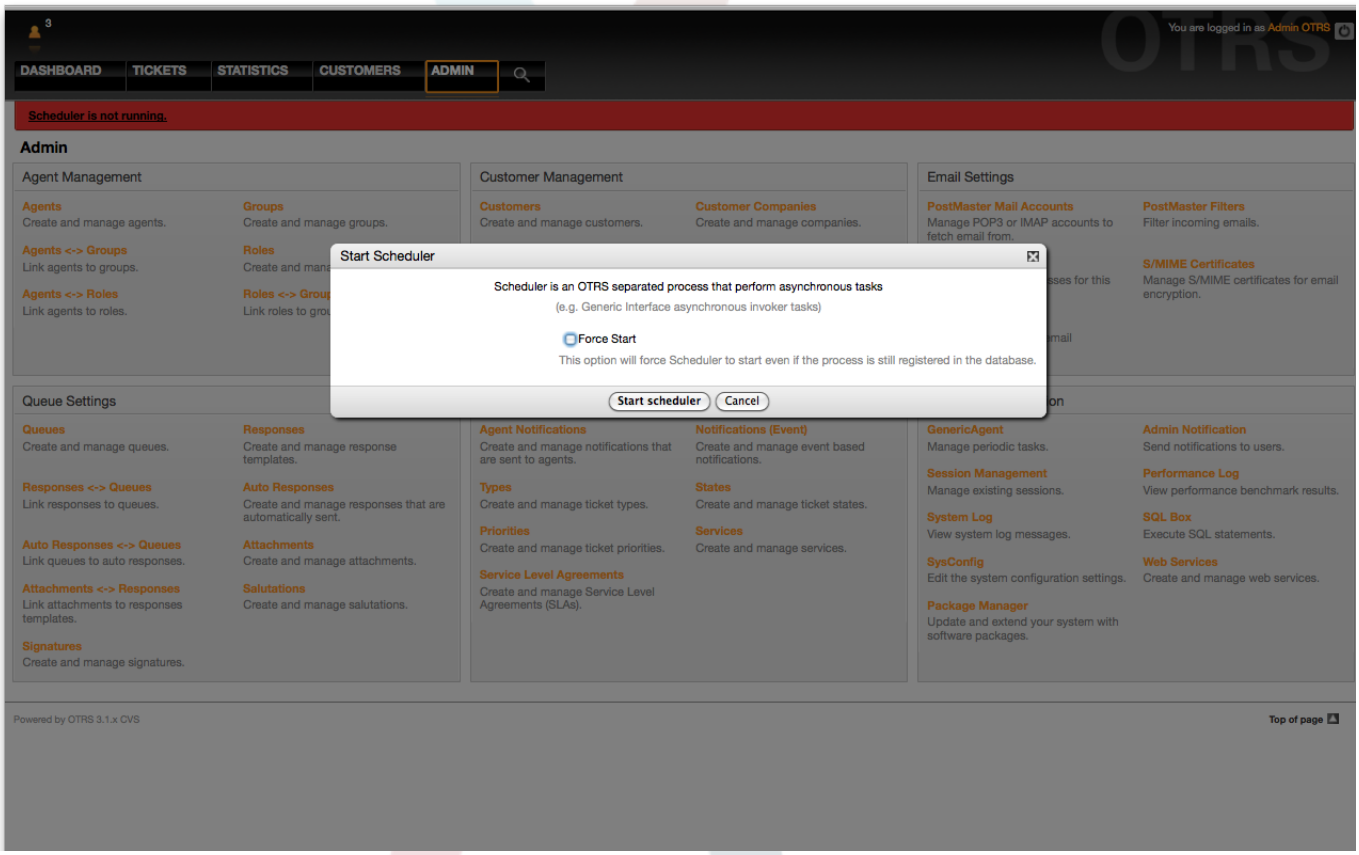
1.2. Start Scheduler

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 it real status please use the command line tools described below.



2. Scheduler Command Line Interface

The Scheduler command line tools let you control the Scheduler process (Start / Stop) or query it status. There are also tools to register the process to be controlled by the operating system.

Included with OTRS there are two set of CLI tools, one for Unix / Linux OS and another for MS Windows OS.

2.1. Unix / Linux

2.1.1. 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 provide init.d scripts to start / stop the OTRS Scheduler process automatically by the operating system, this scripts are located under OTRS_HOME/scripts.

Init.d scripts needs to be copied to the correct location for your operating system, they need to had the proper permissions and some internal variables needs 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
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.

2.1.2. 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.

2.2. Windows

2.2.1. 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 Widows 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.

2.2.2. 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

1. 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 where commonly used in OTRS 3.0 and before. The limitation of these "Free Fields" was that they can be defined up to 16 (text or dropdown) fields and 6 time fields for a ticket and 3 (text or dropdown) fields for each article only, not more.

Now with dynamic fields the limitation in the number of fields per ticket or article is removed, you can create as many dynamic fields you like either for ticket or articles. And beyond that, the framework behind the dynamic fields is prepared to handle custom fields for other objects rather than just ticket and articles.

This new framework that handles the dynamic fields is build using a modular approach, where each kind of dynamic field can be seen as a plug-in module for the framework. This means that the variety of dynamic fields can be easily extended by public OTRS modules, OTRS Feature Add-ons, OTRS custom developments, and other custom developments.

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

2. 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

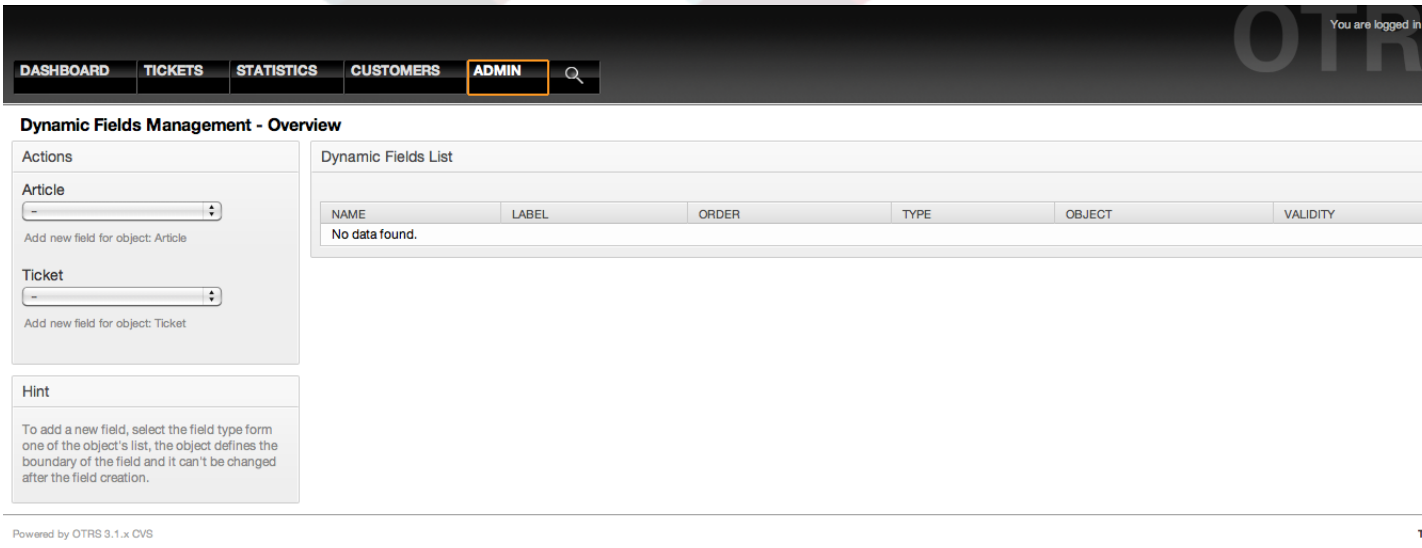
During the migration procedure from OTRS 3.0.x to OTRS 3.1.x all the old "free fields" data and configuration are migrated to the new dynamic fields architecture. Any custom development around the old "free fields" has to be updated to use the new dynamic field framework.

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.

2.1. 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:



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

2.2. 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).

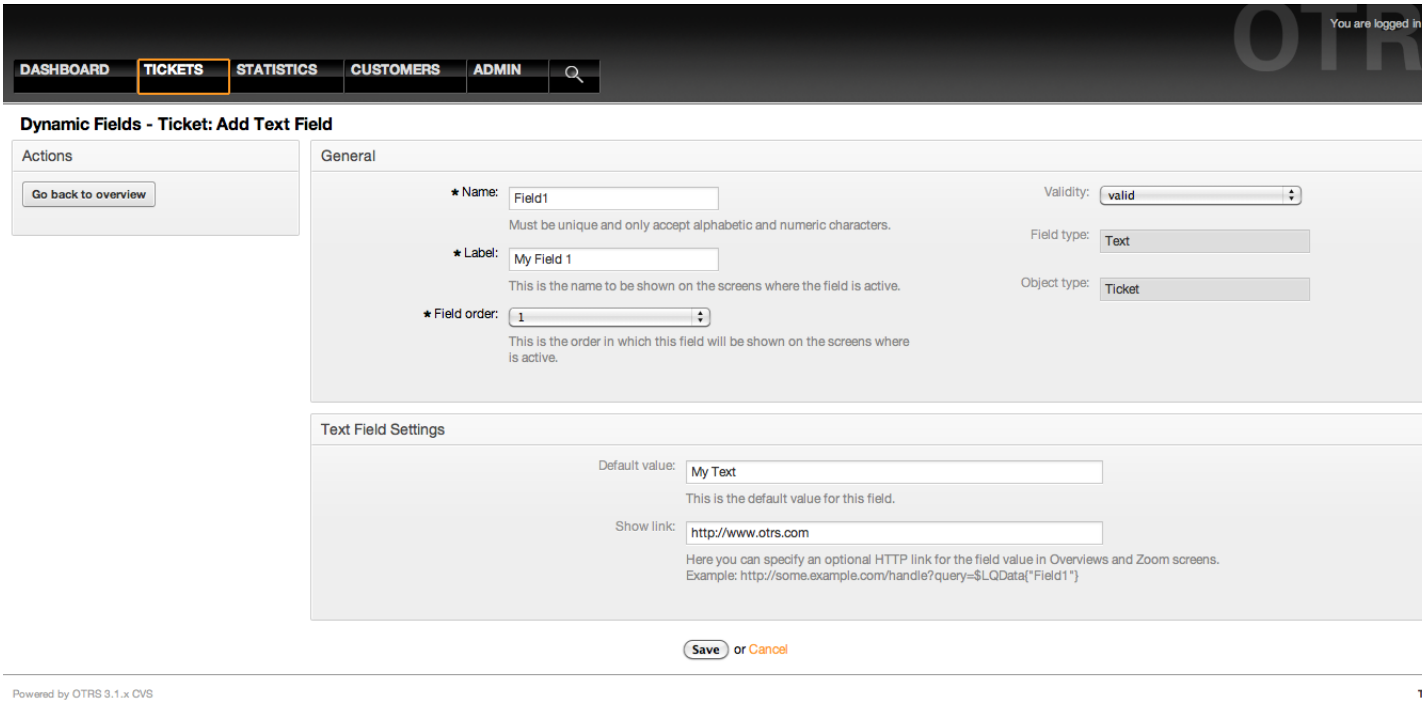
- Show link: Optional.

  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.



2.3. Textarea Dynamic Field Configuration

Textarea dynamic field is used to store a multiple line string.

Textarea dynamic field settings:

- Number of rows: Optional, integer.

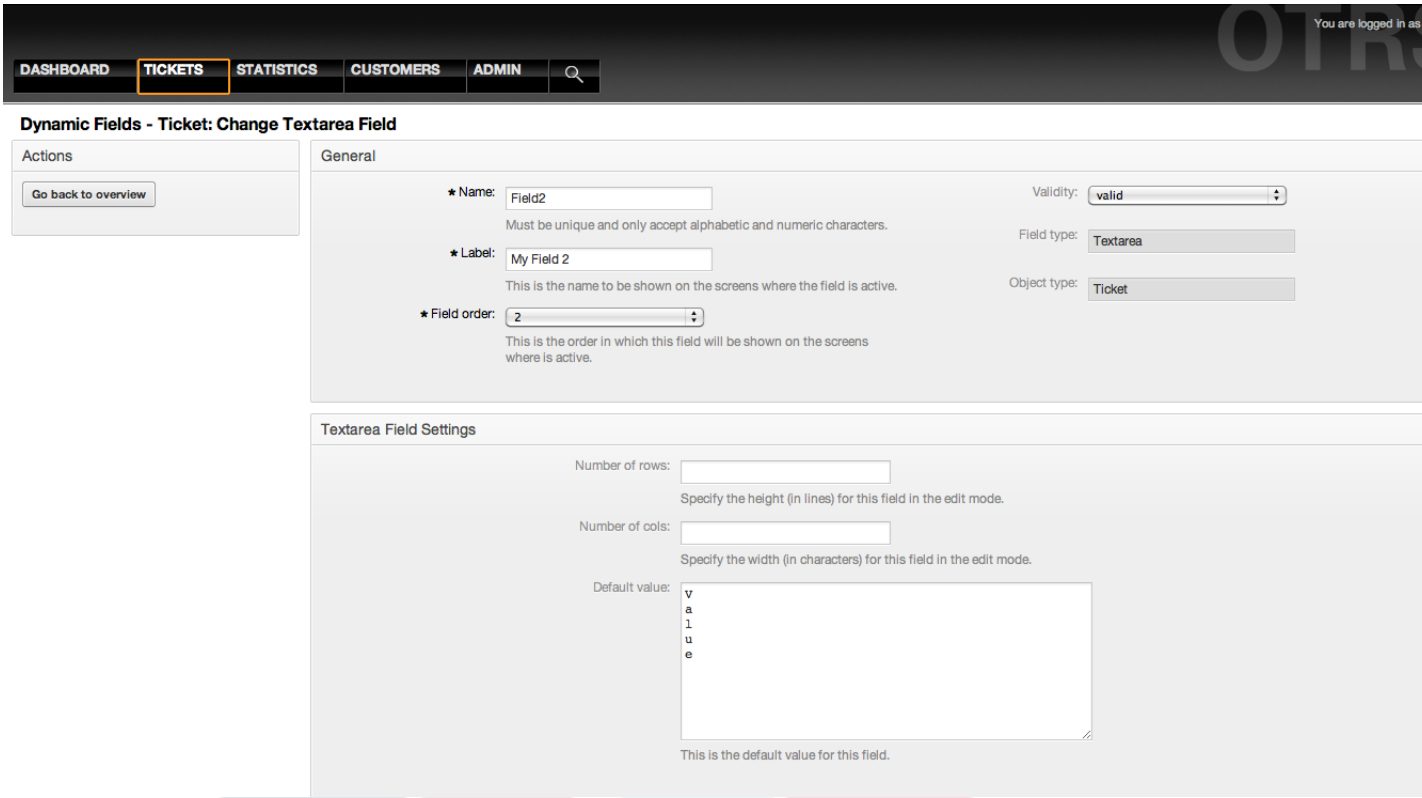
  Used to define the height of the field in the edit screens (like New Phone Ticket or Ticket Compose).

- 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).



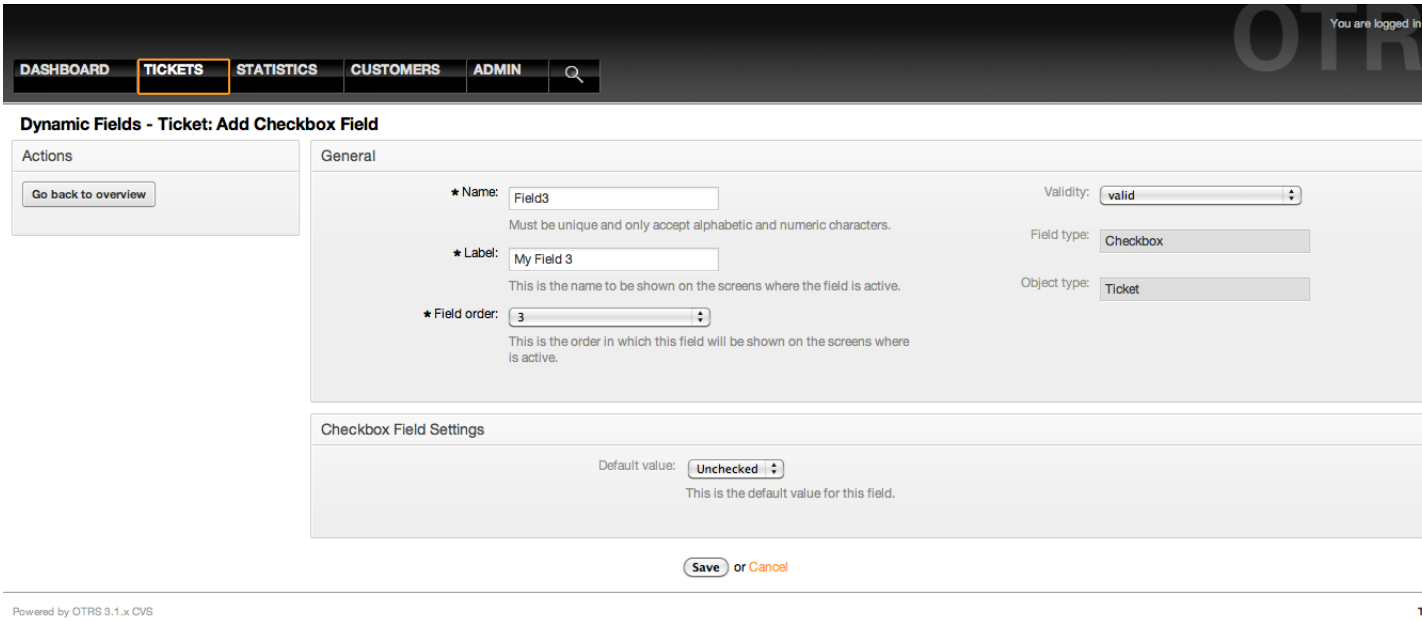
2.4. Checkbox Dynamic Field Configuration

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.



2.5. 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.

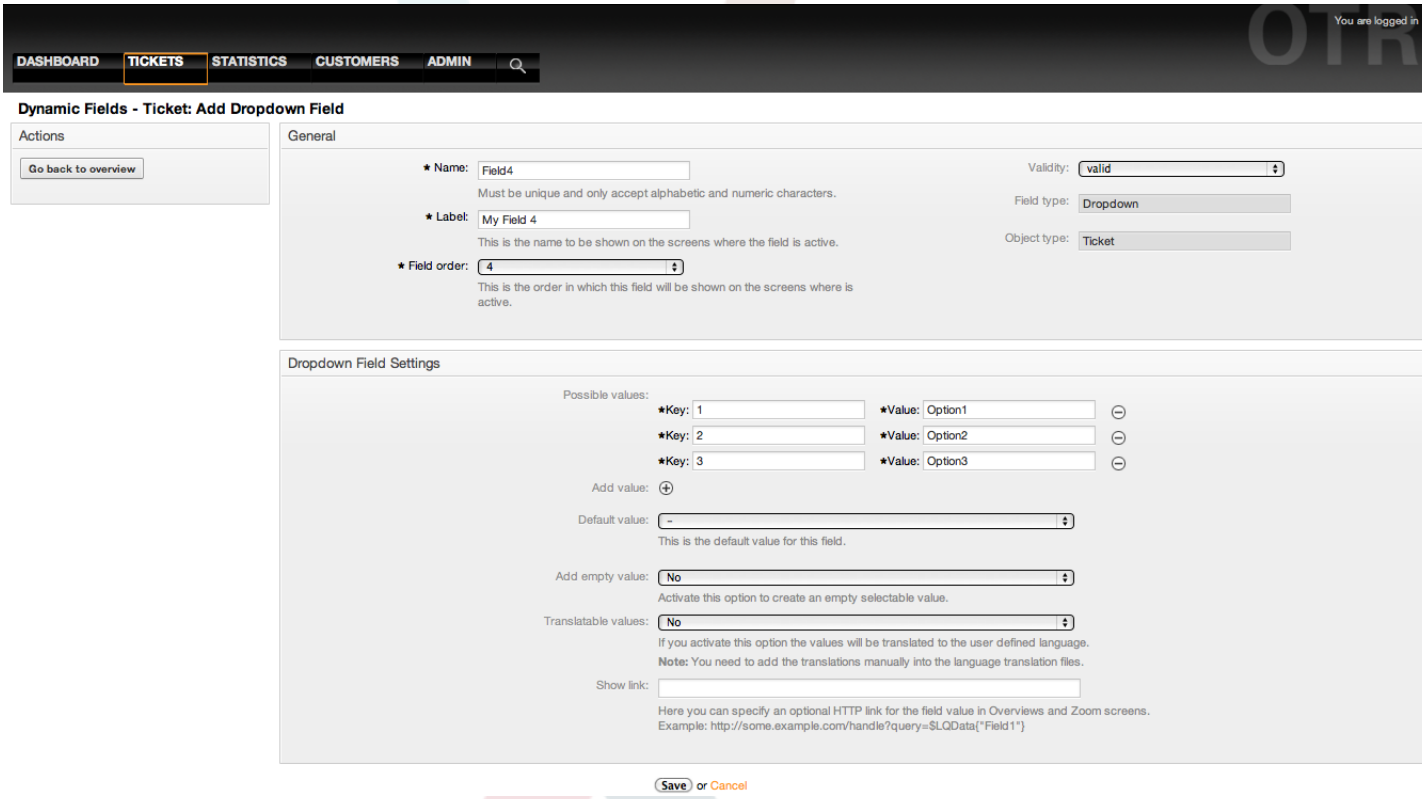
- Show link: Optional.

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.



2.6. 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.

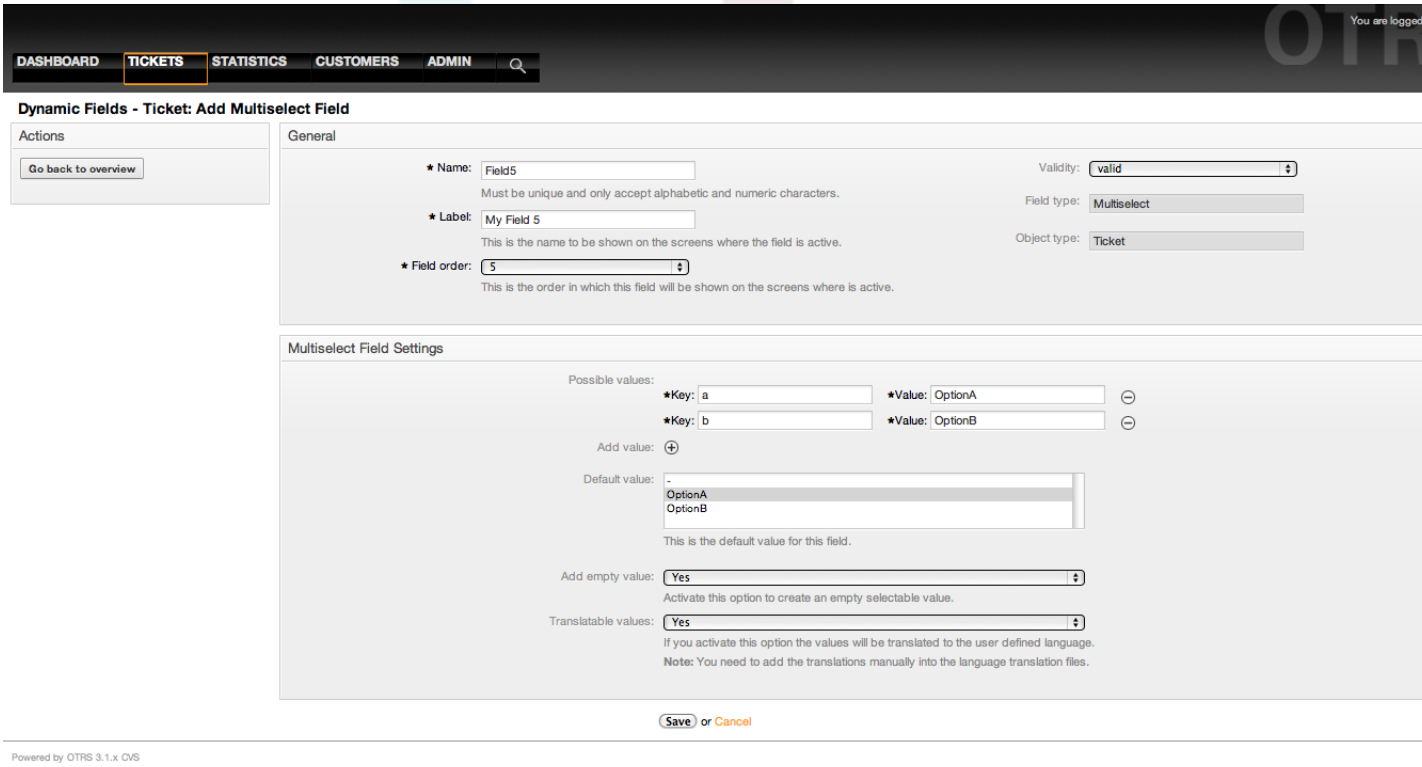
  This is the value to be show by default in the edit screens (like New Phone Ticket or Ticket Compose), the default value for this field is closed selection defined by the Possible values.

- 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.



2.7. 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 dined 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 dined in edit screens.

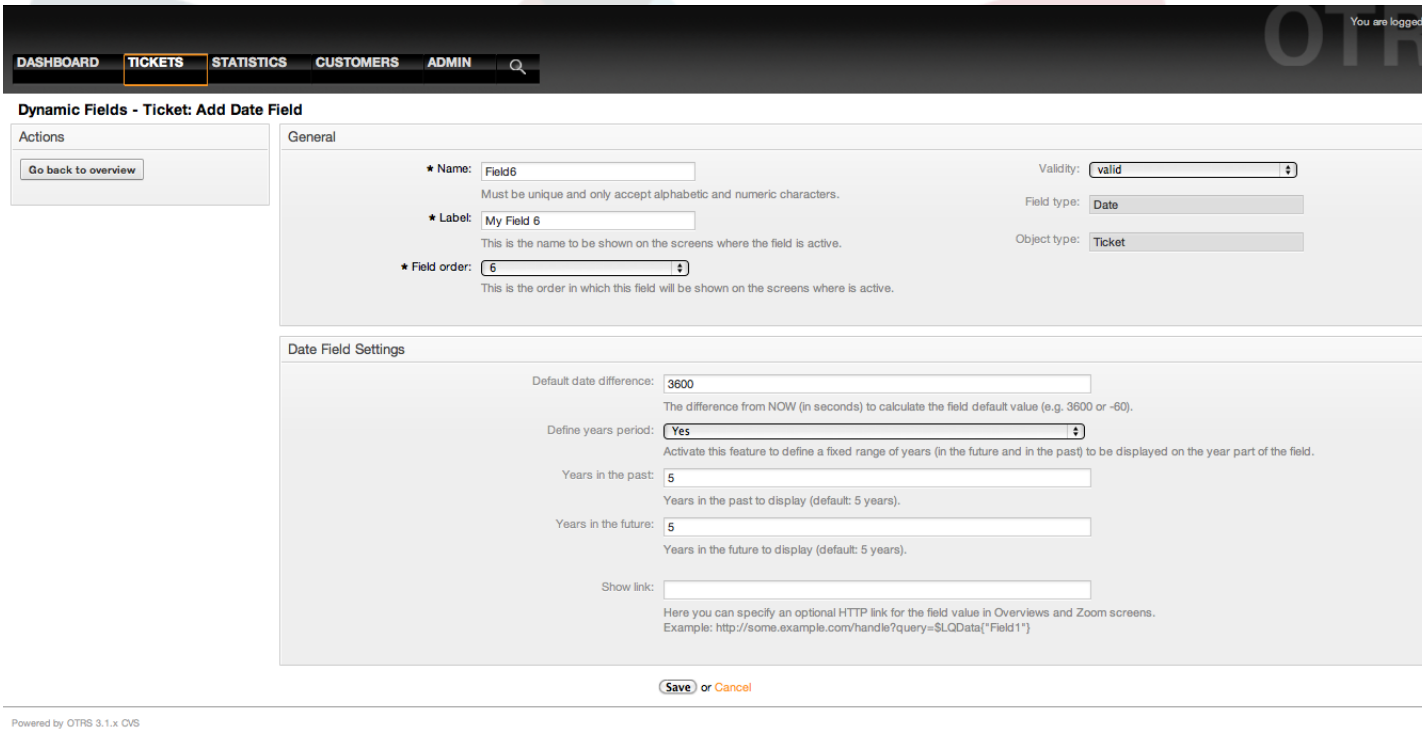
- Show link: Optional.

  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.



2.8. 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 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 dined 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 dined in edit screens.

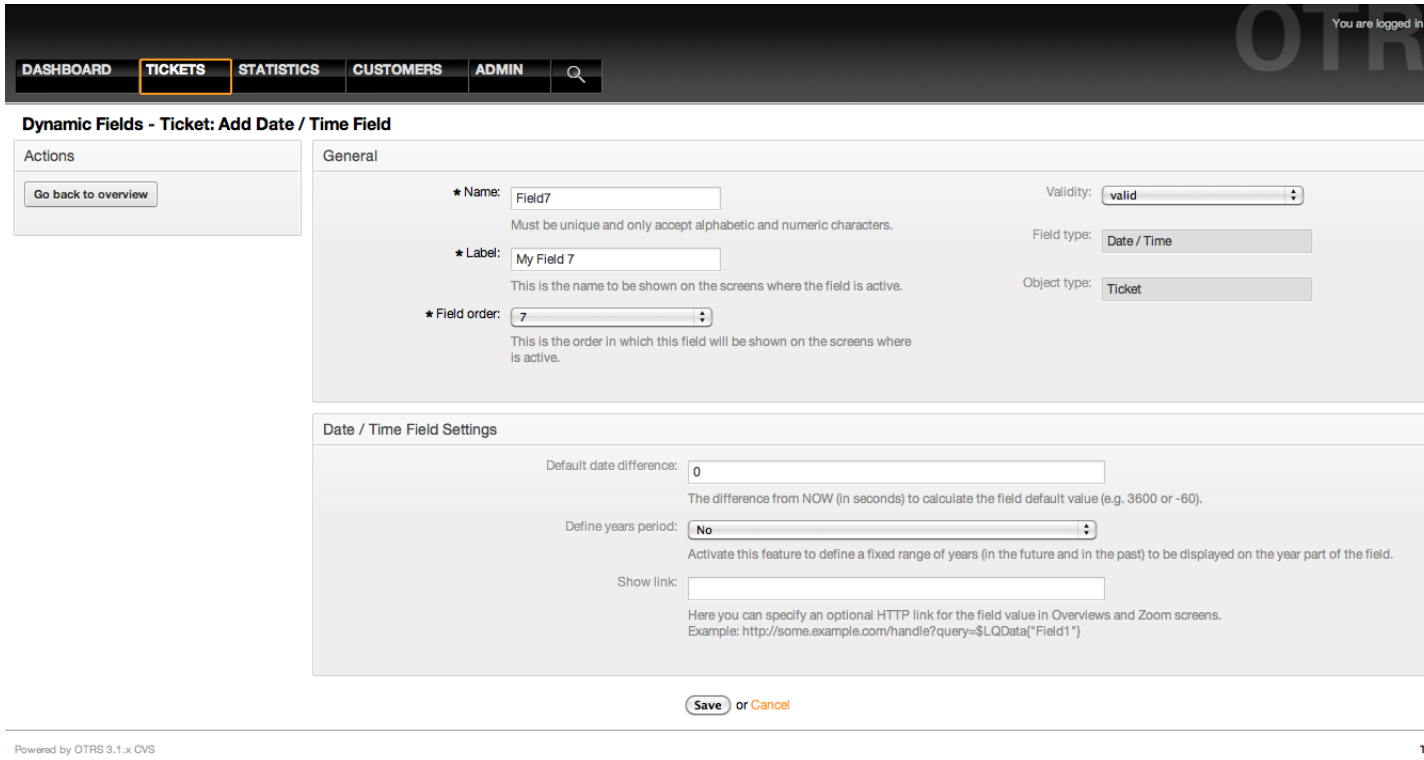
- Show link: Optional.

  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.

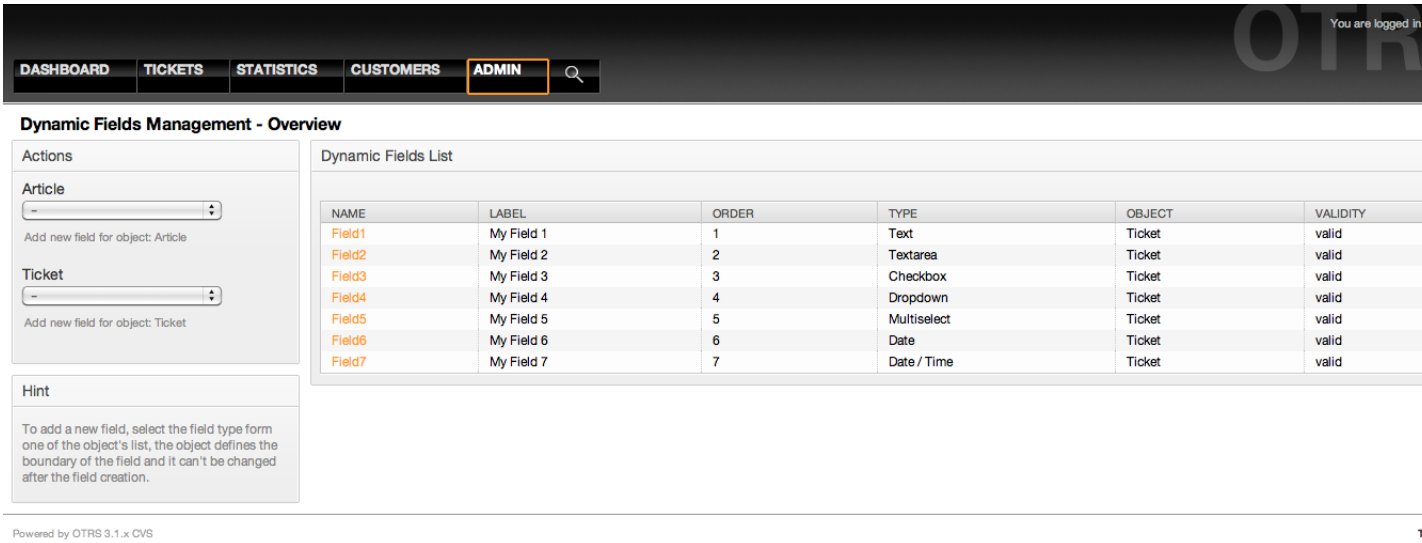
  ### Anmerkung

  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.



2.9. Editing a Dynamic Field

A filled dynamic field overview screen (with the previous examples) should look like:



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).

2.10. 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 field name, click on the "+" button to add a new entry. For example Key: Field1, Content: 1.

  If the setting already has the dynamic field name listed be sure that is set to "1" to display the field or to "2" to display it as mandatory.
- Save the configuration by clicking in the "Update" button and the bottom of the screen and navigate to the screen where you want the field to be displayed.

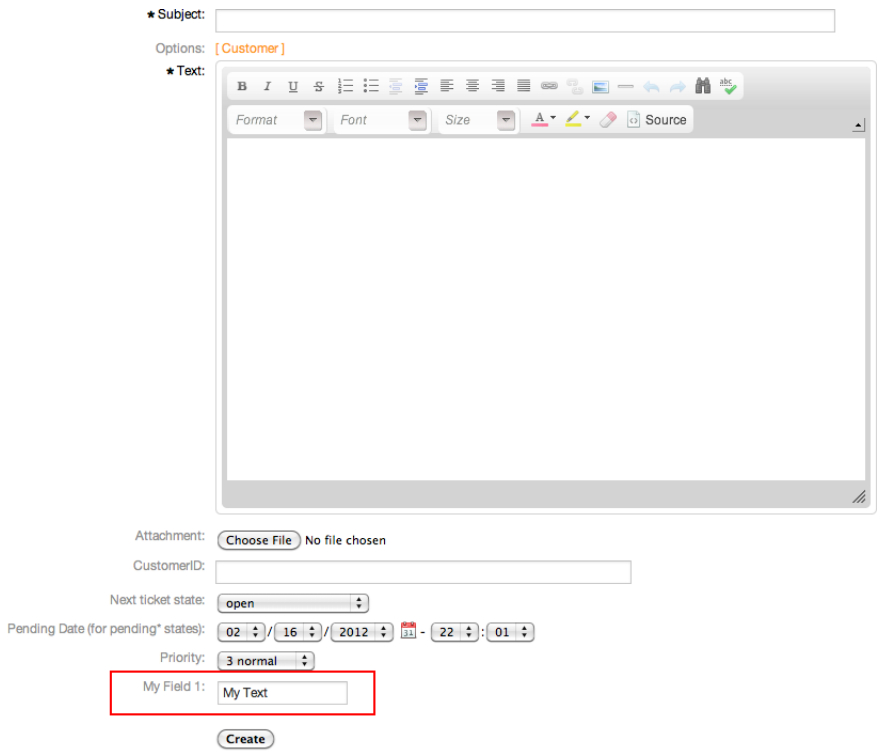
2.10.1. Show Examples

The following are "sysconfig" configurations examples to show or hide dynamic fields on different screens.

Beispiel 22.1. Activate Field1 in New Phone Ticket Screen.

- ⚙⚙⚙⚙⚙⚙ Ticket
- ⚙⚙⚙⚙⚙⚙⚙⚙⚙⚙ Frontend::Agent::Ticket::ViewPhoneNew
- ⚙⚙⚙⚙⚙⚙⚙⚙ Ticket::Frontend::AgentTicketPhone###DynamicField
- ⚙⚙⚙⚙⚙⚙

Key	Content
Field1	1



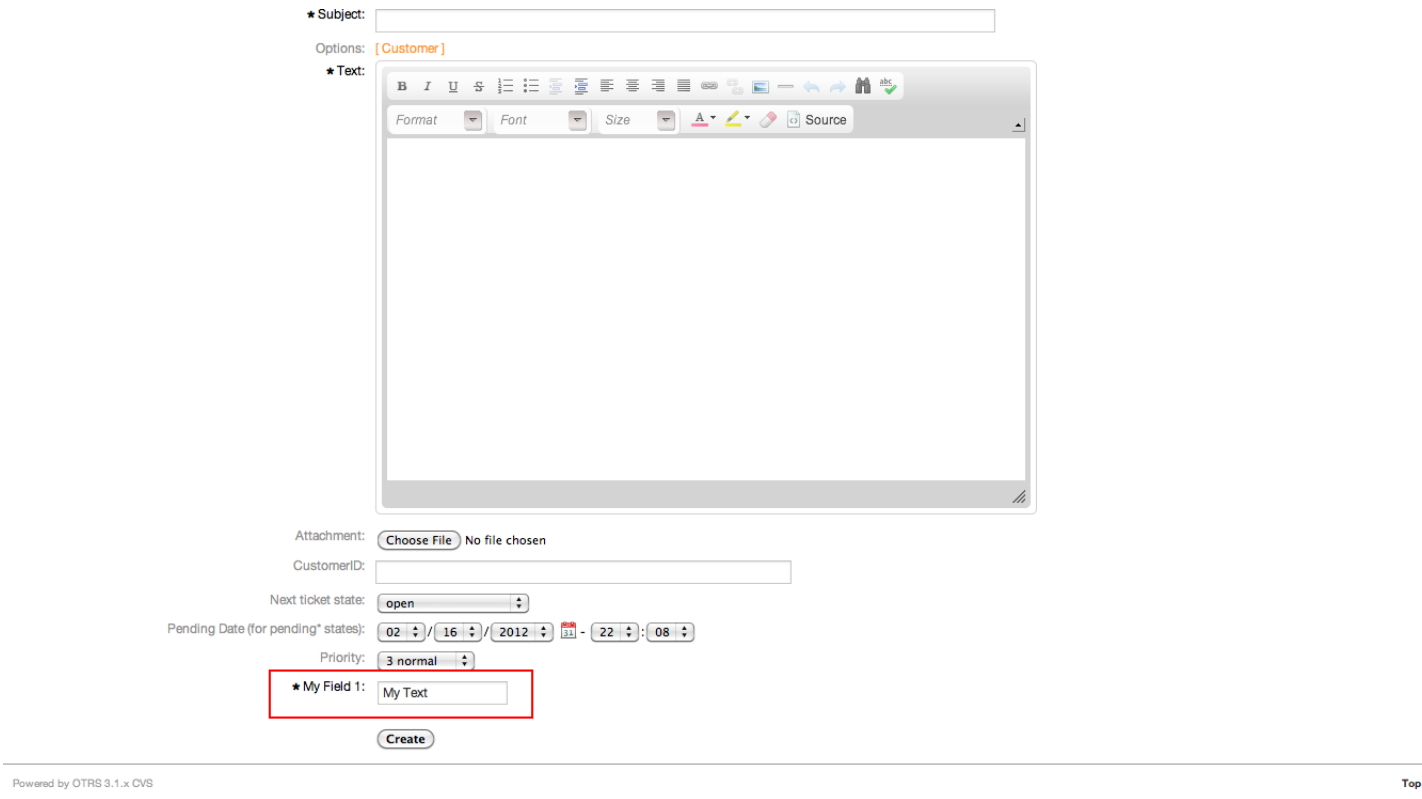
Powered by OTRS 3.1.x CVS

Top

Beispiel 22.2. Activate Field1 in New Phone Ticket Screen as mandatory.

- Ticket
- Frontend::Agent::Ticket::ViewPhoneNew
- Ticket::Frontend::AgentTicketPhone###DynamicField
-

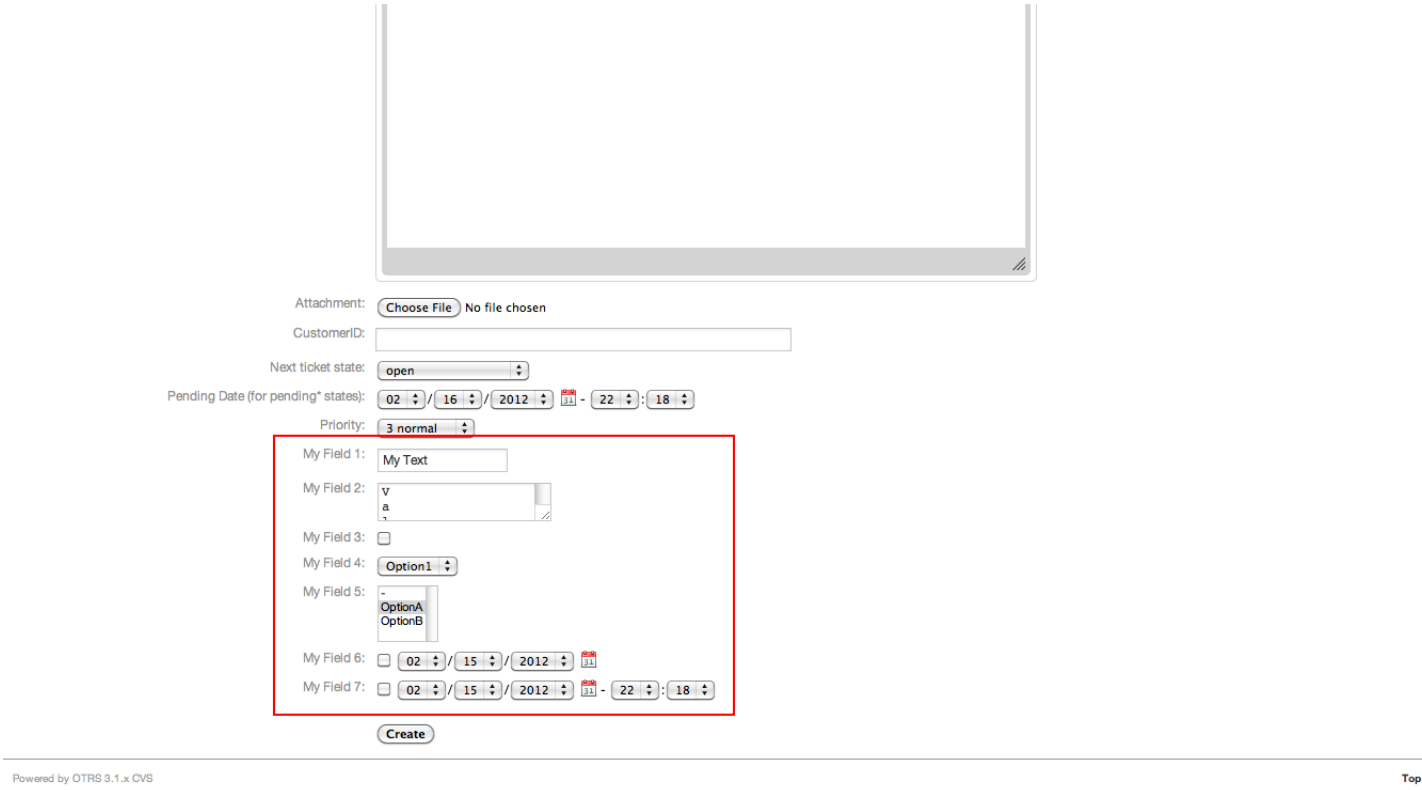
Key	Content
Field1	2



Beispiel 22.3. Activate several fields in New Phone Ticket Screen.

- Ticket
- Frontend::Agent::Ticket::ViewPhoneNew
- Ticket::Frontend::AgentTicketPhone###DynamicField
-

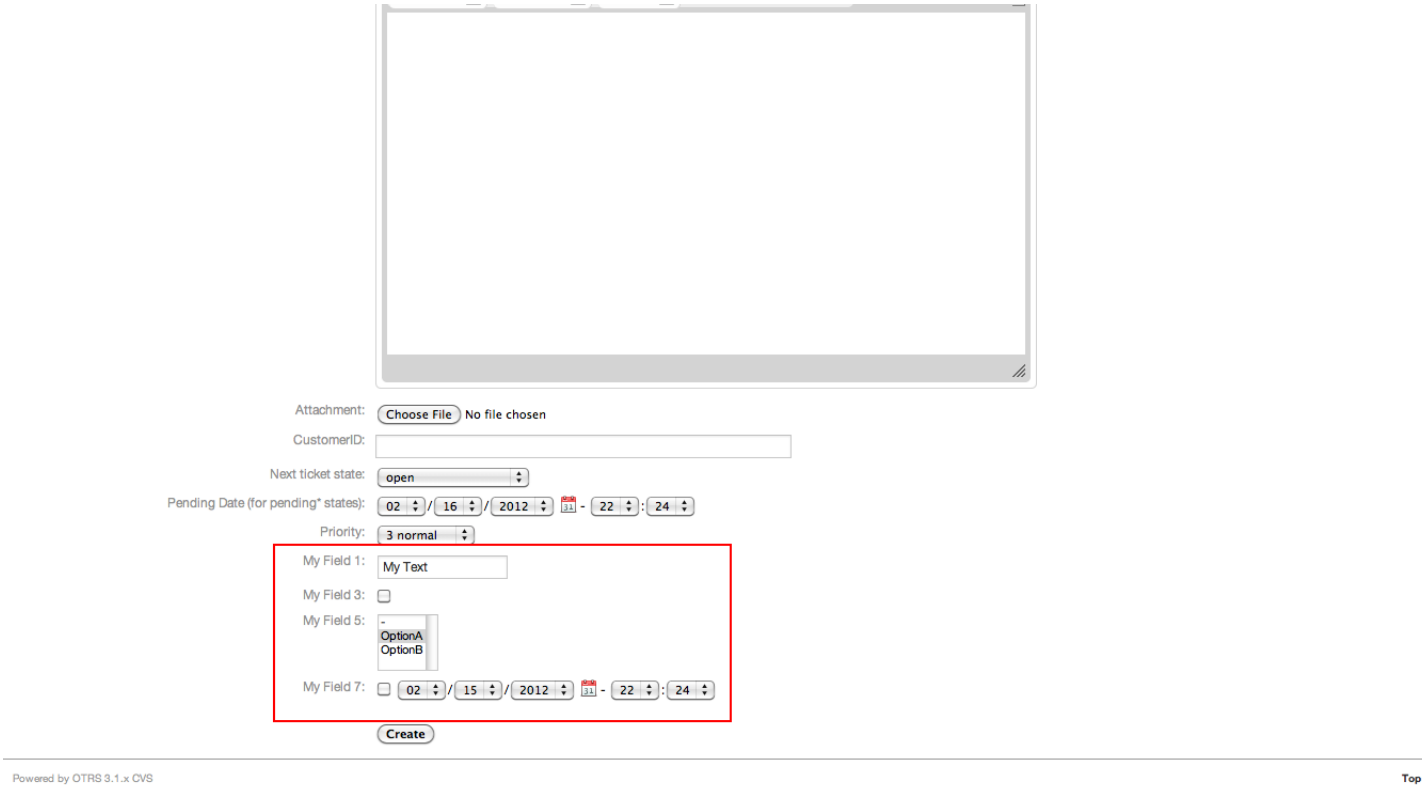
Key	Content
Field1	1
Field2	1
Field3	1
Field4	1
Field5	1
Field6	1
Field7	1



Beispiel 22.4. Deactivate some fields in New Phone Ticket Screen.

- Ticket
- Frontend::Agent::Ticket::ViewPhoneNew
- Ticket::Frontend::AgentTicketPhone###DynamicField
-

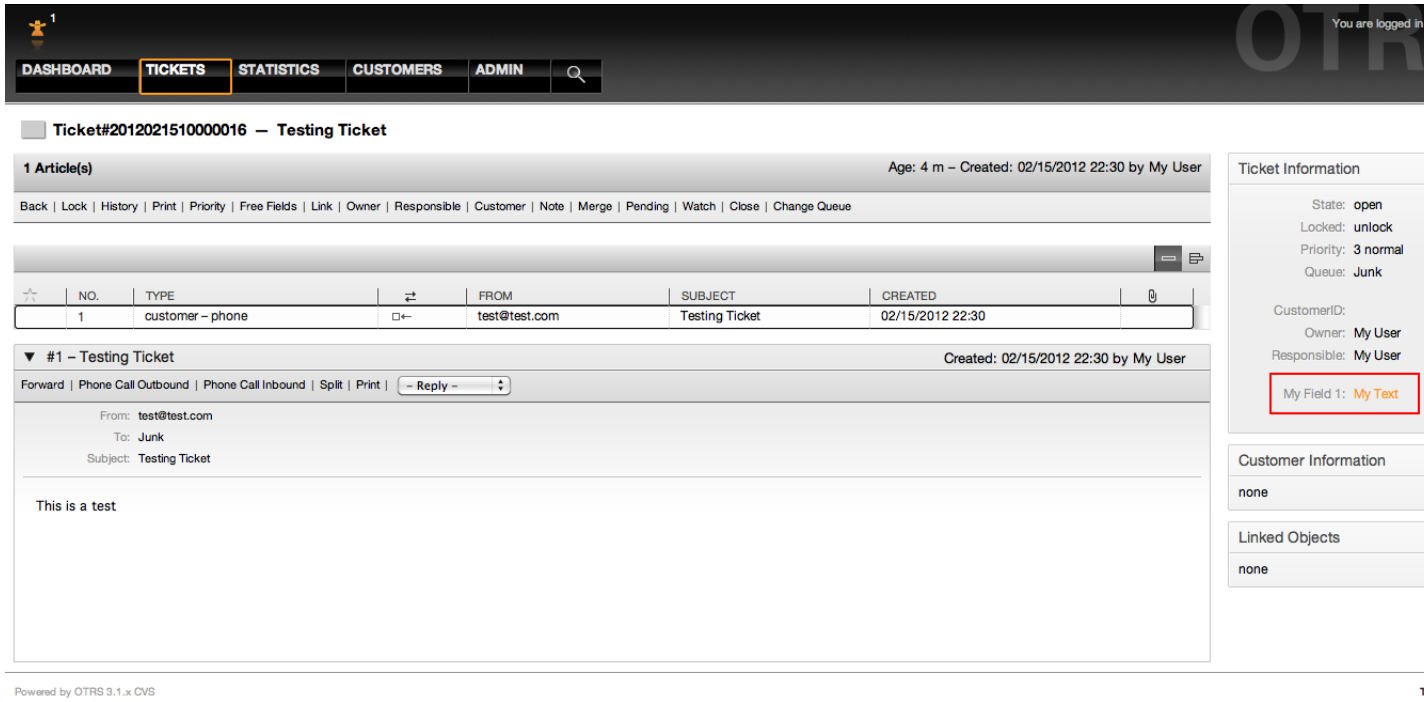
Key	Content
Field1	1
Field2	0
Field3	1
Field4	0
Field5	1
Field6	0
Field7	1



Beispiel 22.5. Activate Field1 in Ticket Zoom Screen.

- Ticket
- Frontend::Agent::Ticket::ViewZoom
- Ticket::Frontend::AgentTicketZoom###DynamicField
-

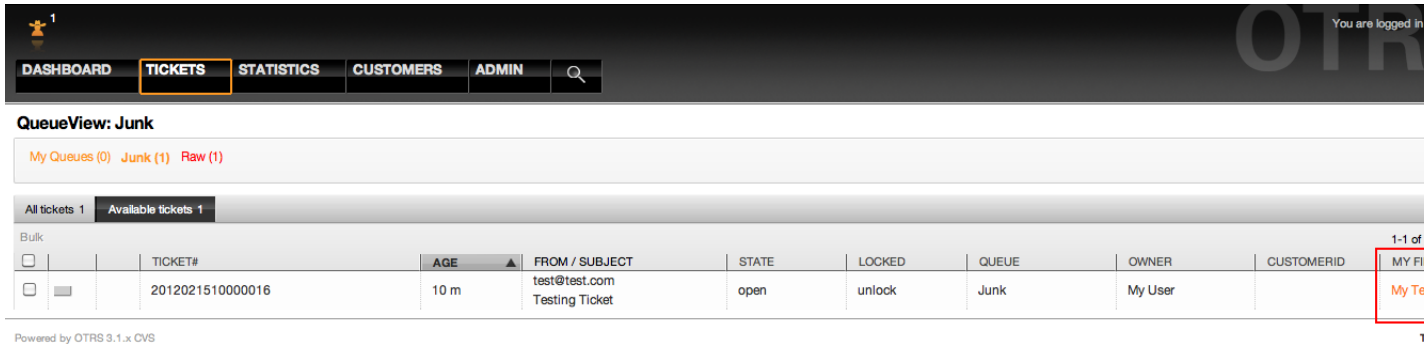
Key	Content
Field1	1



Beispiel 22.6. Activate Field1 in Ticket Overview Small Screens.

- Ticket
- Frontend::Agent::TicketOverview
- Ticket::Frontend::OverviewSmall###DynamicField
-

Key	Content
Field1	1



This setting affects: Escalation View, Locked View, Queue View, Responsible View, Status View and Watch View screens.

2.11. Setting a Default Value by a Ticket Event Module

A ticket event (e.g. TicketCreate) can trigger a value set for a certain field, if the field does not have a value yet.

> **Anmerkung**
>
> By using this method this default value, is not seen in the edit screen (e.g. New Phone Ticket) since the value is set after the creation of the ticket.

To activate this feature is necessary to enable the following setting: "Ticket::EventModulePost###TicketDynamicFieldDefault".

Beispiel 22.7. Activate Field1 in TicketCreate event.

- Ticket
- Core::TicketDynamicFieldDefault
- Ticket::TicketDynamicFieldDefault###Element1

> **Anmerkung**
>
> This configuration can be set in any of the 16 Ticket::TicketDynamicFieldDefault###Element settings.

If more that 16 fields needs to be set up a custom XML file must be places in $OTRS_HOME/Kernel/Config/files directory to extend this feature.

- ⁓⁓⁓⁓⁓⁓

Key	Content
Event	TicketCreate
Name	Field1
Value	a new value

2.12. Set a Default Value by User Preferences

The dynamic field default value can be overwritten with a user defined value stored in the user preferences.

Using this method, the default value of the field will be shown on any screen where the field is activated (if the field does not have already a different value).

The "sysconfig" setting "PreferencesGroups###DynamicField" located in the "Frontend::Agent::Preferences" Sub-group. This setting is an example of how to create an entry in the User Preferences screen to set an exclusive dynamic field default value for the selected user. The limitation of this setting is that it only admits one dynamic field. if two or more fields will use this feature is necessary to create a custom XML configuration file to add more settings similar to this one.

Anmerkung

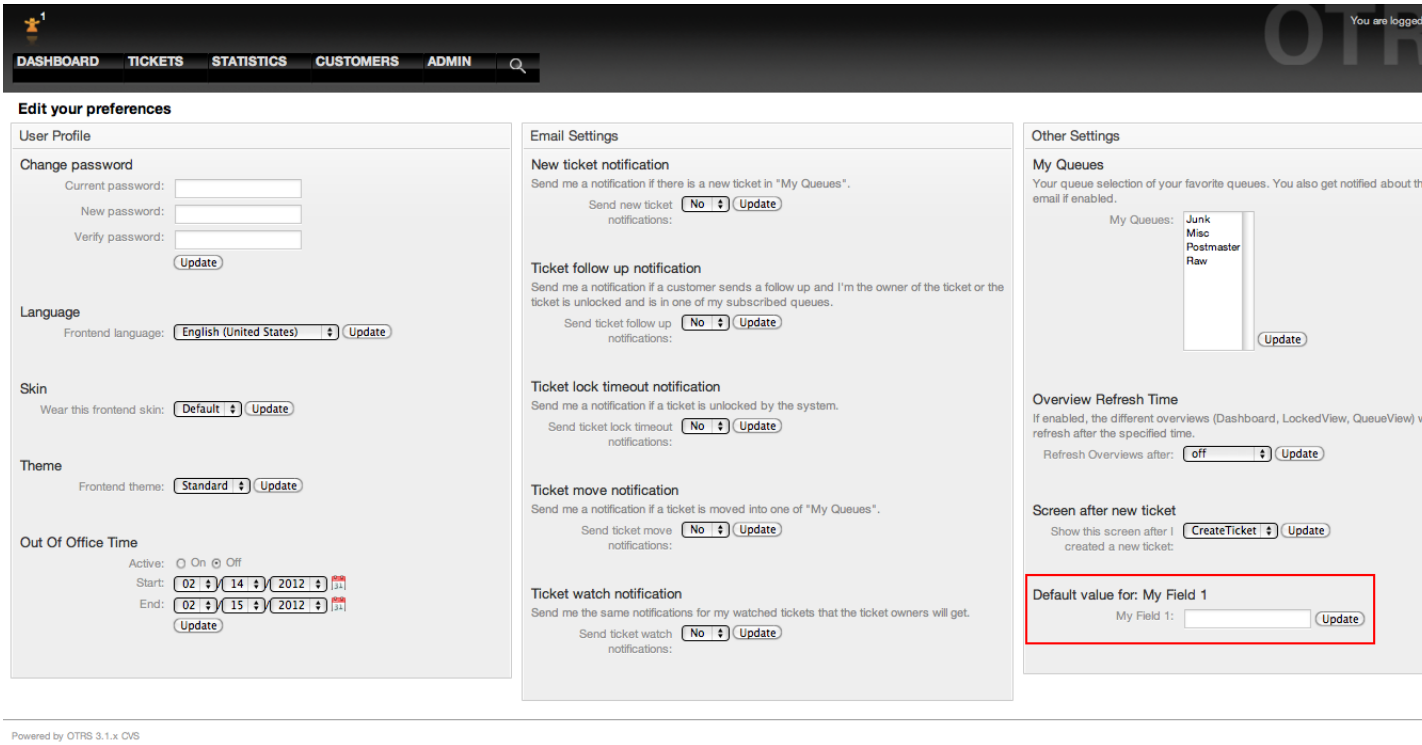
Remember, if more settings are added in a new XML each setting name needs to be unique in the system and different than "PreferencesGroups###DynamicField". for example: PreferencesGroups###101-DynamicField-Field1, PreferencesGroups###102-DynamicField-Field2, PreferencesGroups###My-Field1, PreferencesGroups###My-Field2, etc.

Beispiel 22.8. Activate Field1 in the User preferences.

- ⁓⁓⁓⁓⁓⁓ Ticket
- ⁓⁓⁓⁓⁓⁓⁓⁓⁓⁓ Frontend::Agent::Preferences
- ⁓⁓⁓⁓⁓⁓⁓⁓ PreferencesGroups###101-DynamicField-Field1
- ⁓⁓⁓⁓⁓⁓

Key	Content
Event	TicketCreate
Active	1
Block	Input
Column	Andere Einstellungen
Data:	$Env{"UserDynamicField_Field1"}
Key:	My Field 1
Label:	Default value for: My Field 1

Key	Content
Module:	Kernel::Output::HTML::PreferencesGeneric
PrefKey:	UserDynamicField_Field1
Prio:	7000



2.13. Updating from OTRS 3.0

During the update from OTRS 3.0 there is an special step that is very important and must not be skipped, there is an script file named "DBUpdate-to-3.1.pl" under the "$OTRS_HOME/scripts" that must be called at the precise time in the update procedure (please refer to UPGRADING file for more details).

The script mentioned above is in charge to get all the "Free fields" configurations and creates their dynamic field substitute, it also gathers each "Free field" data from every ticket and article and move it to the dynamic fields format. It also read the screen configuration for "Free fields" and port it to the dynamic fields format.

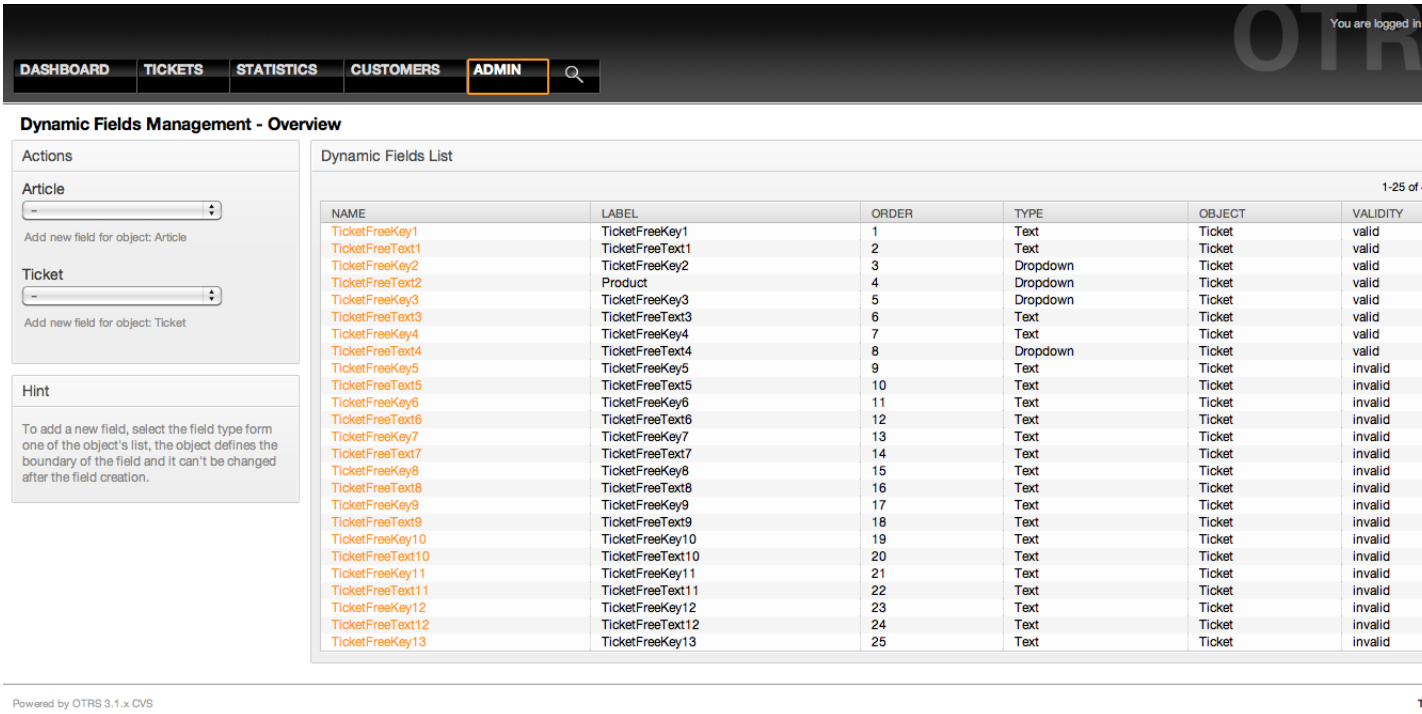
Anmerkung

All the "Free fields" configurations are keep in the system but they are not used anymore, they are placed under the group "Z_Deprecated", and the might be removed in further versions of OTRS.

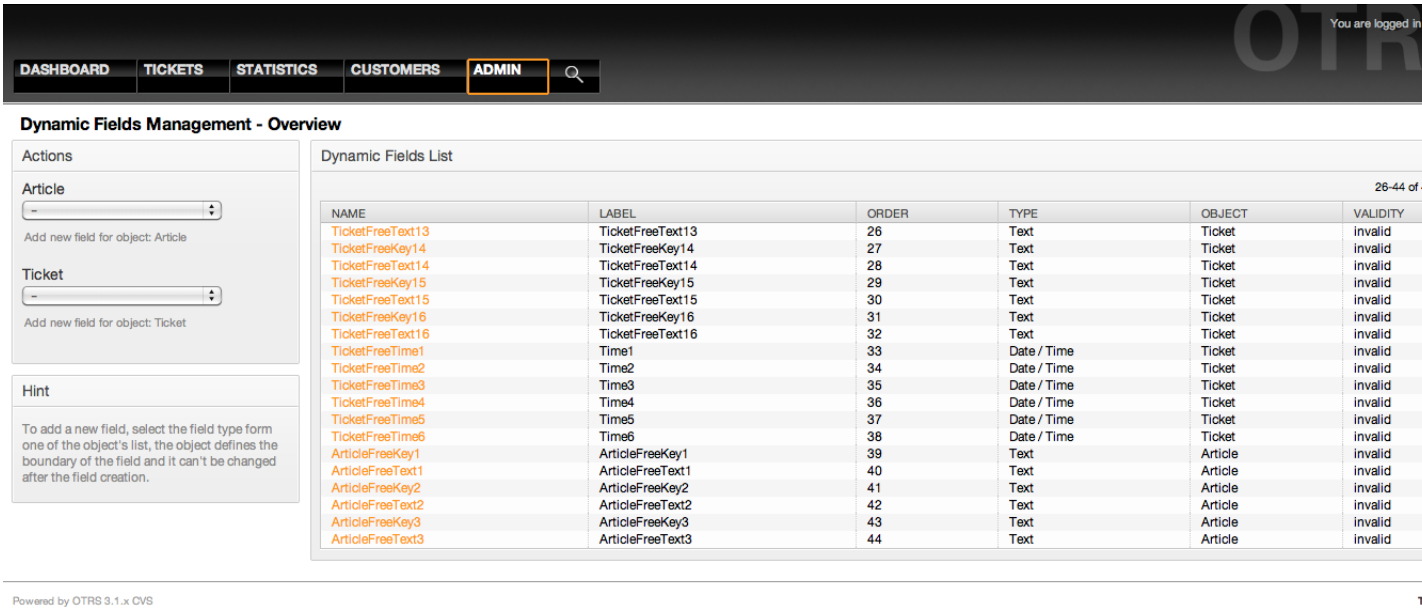
While the updating script tries to be very precise migrating the screen configurations it is always good to do a manual fine tuning. Please note that "Free fields"

TicketFreeKey[1-16] and TicketFreeText[1-16] came always in pair in OTRS 3.0 systems (and before). This is not longer necessary it could be that the TicketFreeKey field is not necessary anymore for a particular TicketFreeText field.

The dynamic fields overview screen from clean OTRS 3.0.x updated to 3.1.x should look like:



NAME	LABEL	ORDER	TYPE	OBJECT	VALIDITY
TicketFreeKey1	TicketFreeKey1	1	Text	Ticket	valid
TicketFreeText1	TicketFreeText1	2	Text	Ticket	valid
TicketFreeKey2	TicketFreeKey2	3	Dropdown	Ticket	valid
TicketFreeText2	Product	4	Dropdown	Ticket	valid
TicketFreeKey3	TicketFreeKey3	5	Dropdown	Ticket	valid
TicketFreeText3	TicketFreeText3	6	Text	Ticket	valid
TicketFreeKey4	TicketFreeKey4	7	Text	Ticket	valid
TicketFreeText4	TicketFreeText4	8	Dropdown	Ticket	valid
TicketFreeKey5	TicketFreeKey5	9	Text	Ticket	invalid
TicketFreeText5	TicketFreeText5	10	Text	Ticket	invalid
TicketFreeKey6	TicketFreeKey6	11	Text	Ticket	invalid
TicketFreeText6	TicketFreeText6	12	Text	Ticket	invalid
TicketFreeKey7	TicketFreeKey7	13	Text	Ticket	invalid
TicketFreeText7	TicketFreeText7	14	Text	Ticket	invalid
TicketFreeKey8	TicketFreeKey8	15	Text	Ticket	invalid
TicketFreeText8	TicketFreeText8	16	Text	Ticket	invalid
TicketFreeKey9	TicketFreeKey9	17	Text	Ticket	invalid
TicketFreeText9	TicketFreeText9	18	Text	Ticket	invalid
TicketFreeKey10	TicketFreeKey10	19	Text	Ticket	invalid
TicketFreeText10	TicketFreeText10	20	Text	Ticket	invalid
TicketFreeKey11	TicketFreeKey11	21	Text	Ticket	invalid
TicketFreeText11	TicketFreeText11	22	Text	Ticket	invalid
TicketFreeKey12	TicketFreeKey12	23	Text	Ticket	invalid
TicketFreeText12	TicketFreeText12	24	Text	Ticket	invalid
TicketFreeKey13	TicketFreeKey13	25	Text	Ticket	invalid



NAME	LABEL	ORDER	TYPE	OBJECT	VALIDITY
TicketFreeText13	TicketFreeText13	26	Text	Ticket	invalid
TicketFreeKey14	TicketFreeKey14	27	Text	Ticket	invalid
TicketFreeText14	TicketFreeText14	28	Text	Ticket	invalid
TicketFreeKey15	TicketFreeKey15	29	Text	Ticket	invalid
TicketFreeText15	TicketFreeText15	30	Text	Ticket	invalid
TicketFreeKey16	TicketFreeKey16	31	Text	Ticket	invalid
TicketFreeText16	TicketFreeText16	32	Text	Ticket	invalid
TicketFreeTime1	Time1	33	Date / Time	Ticket	invalid
TicketFreeTime2	Time2	34	Date / Time	Ticket	invalid
TicketFreeTime3	Time3	35	Date / Time	Ticket	invalid
TicketFreeTime4	Time4	36	Date / Time	Ticket	invalid
TicketFreeTime5	Time5	37	Date / Time	Ticket	invalid
TicketFreeTime6	Time6	38	Date / Time	Ticket	invalid
ArticleFreeKey1	ArticleFreeKey1	39	Text	Article	invalid
ArticleFreeText1	ArticleFreeText1	40	Text	Article	invalid
ArticleFreeKey2	ArticleFreeKey2	41	Text	Article	invalid
ArticleFreeText2	ArticleFreeText2	42	Text	Article	invalid
ArticleFreeKey3	ArticleFreeKey3	43	Text	Article	invalid
ArticleFreeText3	ArticleFreeText3	44	Text	Article	invalid

Please note that the new dynamic field named "TicketFreeText2" has the label "Product", this is because the default configuration from this "Free field" in OTRS 3.0.x specifies that "TicketFreeKey2" has a fixed value of "Product". in this case the dynamic field "TicketFreeKey2" is not needed anymore and it can be manually set to invalid.

It might me also a very good idea to set the label value of each valid migrated field to a text that describes better the use of the field in the system.

Kapitel 23. Zusätzliche Applikationen

Zusätzlich zum OTRS-Framework haben Sie die Möglichkeit, weitere Applikationen aus einem Online-Verzeichnis über das Admin-Interface (Paketmanager) zu installieren. Dieser lädt Applikationen von einem Online-Verzeichnis herunter und löst die Paketabhängigkeiten auf. Es ist zudem möglich, Pakete in Form lokaler Dateien zu installieren, z. B. für den Fall, dass Ihr OTRS keine Internetverbindung hat.

1. FAQ

Mit Hilfe des FAQ-Systems kann eine Wissensdatenbank aufgebaut und verwaltet werden. Einträge lassen sich nur intern oder auch extern, d.h. für alle Kunden oder komplett öffentlich, freischalten. Einträge können nach Sprache oder nach Kategorien erstellt und sortiert werden.

Kapitel 24. Leistungsverbesserung

Hier finden Sie eine Liste verschiedener Techniken der Leistungssteigerung Ihrer OTRS-Installation, einschließlich Konfiguration, Programmierung, Speichernutzung und mehr.

1. OTRS

Es gibt verschiedene Ansätze zur Leistungssteigerung von OTRS.

1.1. TicketIndexModule

Zur Verfügung stehen zwei Backend-Module für den Ticket Index:

- Kernel::System::Ticket::IndexAccelerator::RuntimeDB (Standard), generiere jede Queue-Ansicht dynamisch aus der Ticket Tabelle. Sie werden keine Probleme mit der Leistung bekommen bis zu etwa 60.000 Tickets (oder 6000 offenen) in Ihrem System.

- Kernel::System::Ticket::IndexAccelerator::StaticDB, das leistungsfähigste Modul. Es sollte ab 80.000 Tickets oder mehr als 6000 offenen eingesetzt werden. Benutzt eine extra ticket_index Tabelle, arbeitet wie eine Ansicht (View). Führen Sie `bin/otrs.RebuildTicketIndex.pl` zum erstmaligen Aufbau des Index aus.

Sie können den IndexAccelerator per SysConfig einstellen.

1.2. TicketStorageModule

Es stehen zwei Module für das Speichern der Tickets und Artikel bereit:

- Kernel::System::Ticket::ArticleStorageDB (Standard), speichert Anhänge u. A. in der Datenbank. Merke: Benutzen Sie diese Option nicht für größere Systeme.

  Pro: Ist der Benutzer, unter dem der Webserver läuft, nicht der Benutzer 'otrs', können Sie mit diesem Modul Dateiberechtigungsprobleme vermeiden.

  Contra: Es ist nicht wirklich ratsa,, Anhänge in Ihrer Datenbank zu speichern. Achten Sie darauf, dass Ihre Datenbank das kann. Für MySQL setzen Sie in dessen Konfiguration bspw. "set-variable = max_allowed_packet=8M", um 8 MB große Objekte zu speichern (Standard ist 2M).

- Kernel::System::Ticket::ArticleStorageFS, speichert Anhänge u. A. im lokalen Filesystem ab. Merke: Benutzen Sie dies für große Installationen.

  Pro: Schneller!

  Contra: Der Benutzer, unter dem der Webserver läuft, sollte der Benutzer 'otrs' sein (Dateisystemberechtigungen!). Wenn Sie mehrere OTRS-Frontendserver haben, müssen Sie sicherstellen, dass das Dateisystem gemeinsam genutzt wird. Sie können es z. B. auf ein NFS-Share oder vorzugsweise ein SAN oder eine vergleichbare Lösung legen.

Merke: Sie können während des Betriebs von einem Backend auf das andere wechseln. Stellen Sie dazu das Backend in der SysConfig um, und führen Sie `otrs.ArticleStorageSwitch.pl` aus, um die Artikel aus der Datenbank ins Dateisystem zu überführen oder umgekehrt. Sie können die `-s` und `-d` Option verwenden, um das Quell- und Zielbackend anzugeben. Der Prozess kann eine ganze Weile dauern, je nach Menge der Artikel sowie System- und Netzwerkleistung.

```
shell> bin/otrs.ArticleStorageSwitch.pl -s ArticleStorageDB -d ArticleStorageFS
```

1.3. Tickets archivieren

Da OTRS als revisionssicheres System betrieben werden kann, ist das Löschen von geschlossenen Tickets möglicherweise nicht empfehlenswert. Daher haben wir eine Funktion implementiert, mit der Sie Tickets archivieren können.

Kokret ist darunter zu verstehen, dass Tickets, die bestimmte Kriterien erfüllen, als "archiviert" markiert werden. Diese Tickets werden dann bei regulären Suchabfragen oder von GenericAgent-Jobs nicht mehr erfasst. Somit muss sich das System mit einer großen Ticketmenge nicht mehr befassen, weil dann nur noch die "aktuellsten" Tickets betrachtet werden. Das kann auf großen Systemen eine signifikante Performanceverbesserung bewirken.

Befolgen Sie folgende Schritte, um die Archivierungsfunktion zu nutzen:

1. Archivsystem in der SysConfig aktivieren

   Wählen Sie in der SysConfig die Gruppe `Ticket` aus. In `Core::Ticket` finden Sie die Option `Ticket::ArchiveSystem`, die standardmäßig auf "Nein" steht. Ändern Sie diese auf "Ja" und speichern Sie die Änderung ab.

2. Anlegen eines GenericAgent-Jobs

   Wählen Sie im Administrationsbereich den "GenericAgent" aus und legen Sie dort einen neuen Job an.

   a. Job-Einstellungen

      Geben Sie dem Job einen geeigneten Namen und angemessene Optionen.

   b. Ticket-Filter

      Der Ticketfilter ist eine Ticketsuche, die Tickets nach bestimmten Kriterien auswählt. Es könnte empfehlenswert sein, nur Tickets zu archivieren, die seit einigen Monaten im Status "geschlossen" sind.

   c. Ticket-Aktion

      Im Abschnitt "Ticket-Aktion" werden Sie eine Aktion "Ausgewählte Tickets archivieren" finden. Wählen Sie dort "Tickets archivieren" aus.

   d. Job speichern

      Am Ende der Seite finden Sie einen Knopf zum Speichern des Jobs.

   e. Betroffene Tickets

      Das System wird dann alle Tickets anzeigen, die beim Ausführen des GenericAgent-Jobs archiviert werden.

3. Ticketsuche

   Wenn Sie nun nach Tickets suchen, werden standardmäßig nur Tickets gefunden, die nicht archiviert sind. Wenn Sie auch in archivierten Tickets suchen wollen, fügen Sie "Archivsuche" zu Ihren Suchkriterien hinzu.

2. Datenbank

Einstellungen sind immer spezifisch für die jeweils eingesetzte Datenbank. Bei Problemen lesen Sie die Dokumentation und fragen Sie Ihren Datenbankadministrator.

2.1. MySQL

Wenn Sie den Tabellentyp MyISAM (Standard) benutzen, und einen großen Teil einer Tabelle gelöscht haben, oder wenn Sie sehr viele Änderungen an einer Tabelle mit Zeilen variabler Länge vorgenommen haben (Tabellen mit VARCHAR, BLOB oder TEXT Spalten), sollten Sie die Datendateien mit dem "optimize" Kommando behandeln.

Dies bietet sich an, wenn MySQL viel CPU Zeit braucht. Optimieren Sie die Tabellen ticket, ticket_history und article.

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

2.2. PostgreSQL

PostgreSQL konfigurieren Sie am besten in der postgresql.conf Datei in Ihrem PostgreSQL Datenverzeichnis. Hier gibt es Hilfe dazu:

-
-
-

Ist die Leistung immer noch nicht genügend, empfehlen wir, Fragen auf der "PostgreSQL Performance Mailing Liste" (http://www.postgresql.org/community/lists/) zu stellen. Die Teilnehmer der PostgreSQL Liste sind sehr freundlich und können wahrscheinlich helfen.

3. Webserver

Natürlich empfehlen wir mod_perl 2.0 (http://perl.apache.org/). Es ist sehr viel schneller (etwa um den Faktor 100) als pures CGI, braucht aber auch mehr Speicher. Ihr httpd wird mit mod_perl also größer sein.

3.1. Persistente Datenbankverbindungen

Sie können die Datenbankverbindung bereits beim Start des Webservers herstellen lassen. Dies spart ebenso Zeit (siehe auch README.webserver).

3.2. Vorgeladene Module - startup.pl

Nutzen Sie das Start Skript `scripts/apache2-perl-startup.pl`, um die Perl Module vorzuladen (siehe README.webserver). Dadurch wird der Webserver schneller und braucht weniger Speicher.

3.3. Perl Module bei Änderung neu laden

By default Apache::Reload is used in scripts/apache2-httpd.include.conf. Disable it and you will get 8% more speed. But remember to restart the web server if you install any modules via the OTRS Package Manager, or any values in your SysConfig or in Kernel/Config.pm. Important: this would also mean you can't use the OTRS Package Manager via the web interface, you need to use the command line variant - `bin/otrs.PackageManager.pl`.

3.4. Die richtige Strategie wählen

If you have a larger installation, say over 1,000 new tickets per day and over 40 agents, it is a good idea to read the chapters on Performance of the mod_perl User's Guide ( ).

3.5. mod_gzip/mod_deflate

Falls Ihre Bandbreite ein wenig schmal sein sollte, benutzen Sie mod_deflate für Apache2. Eine HTML-Seite von 45k wird mod_gzip/mod_deflate auf etwa 7k zusammendrücken. Allerdings wird dadurch die Last auf dem Server erhöht.

Kapitel 25. Datensicherung

In diesem Kapitel wird beschrieben, wie alle relevanten Daten der OTRS-Installation gesichert und wieder hergestellt werden können.

1. Datensicherung

Bei einem Backup gibt es zwei Arten von Datensicherung, die Applikation (z. B. `/opt/otrs/`) und die Datenbank.

Um Backups zu vereinfachen, wird ein `scripts/backup.pl` mitgeliefert, das alle benötigten Komponenten sichert.

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

Ein Backup kann also z. B. mit folgendem Befehl erstellt werden:

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

Alle Daten wurden in das Verzeichnis `/backup/2010-09-12_14-28/` gesichert und dort, getrennt nach Art der Daten, in einzelne .tar.gz-Dateien gespeichert.

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

2. Wiederherstellung der Daten

Um ein Backup wieder einzuspielen, müssen die Applikation (z. B. nach `/opt/otrs/`) und die Datenbank wieder hergestellt werden.

Um Backups einspielen zu können, wird ein `scripts/restore.pl` mitgeliefert, das die benötigten Komponenten wieder zurück sichert. Es unterstützt MySQL und PostgreSQL.

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

Daten, die im Verzeichnis `/backup/2010-09-12_14-28/` gesichert wurden, können bei einer OTRS-Installation in `/opt/otrs` folgendermaßen zurückgespielt werden.

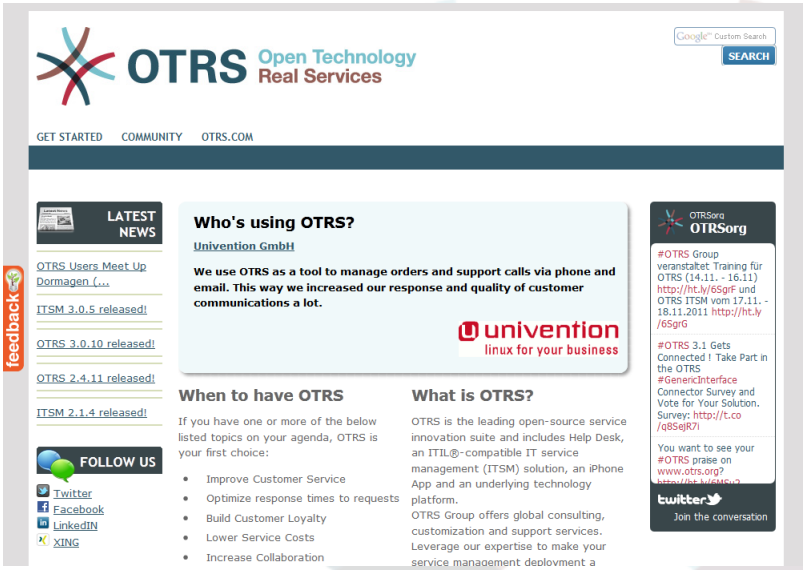
```
linux:/opt/otrs/scripts# ./restore.pl -b /backup/2010-09-07_14-28 -d /opt/otrs/
Restore /backup/2010-09-07_14-28//Config.tar.gz ...
Restore /backup/2010-09-07_14-28//Application.tar.gz ...
create MySQL
decompresses SQL-file ...
cat SQL-file into MySQL database
compress SQL-file...
linux:/opt/otrs/scripts#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

Anhang A. Weitere Quellen

Wir unterstützen Sie mit den aktuellsten Informationen über OTRS und geben Ihnen die Möglichkeit, uns mit Feedback zu versorgen.

1. Homepage OTRS.org

Sie finden unsere Projekt-Homepage unter .



2. Mailinglisten

Die folgende Tabelle gibt einen Überblick darüber, welche Community-Mailinglisten es derzeit für OTRS gibt und wie diese zu erreichen sind:

Tabelle A.1. Mailinglisten

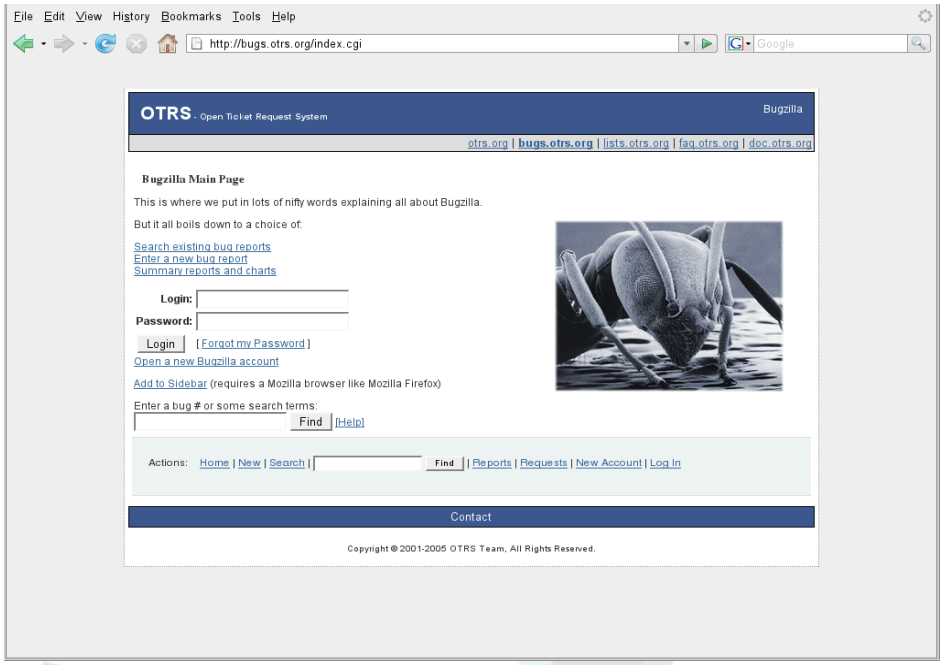
Name	Beschreibung	Homepage
announce@otrs.org	Ankündigungen zu OTRS (neue Versionen, Updates, ...) mit wenig Verkehr.	
otrs@otrs.org	Benutzer-Fragen und -Diskussionen rund um OTRS (englischsprachig) mit mittel bis viel Verkehr.	
Ç	Benutzer-Fragen und -Diskussionen rund um OTRS (deutschsprachig) mit mittel bis viel Verkehr.	

Name	Beschreibung	Homepage
dev@otrs.org	Mailingliste für die Entwickler von OTRS mit wenig Verkehr.	
doc-de@otrs.org	Deutsche Diskussion über die OTRS-Dokumentation mit wenig Verkehr.	
i18n@otrs.org	Englische Mailingliste mit wenig Verkehr für Übersetzungs- und Lokalisierungsfragen. Wenn Sie OTRS übersetzen oder übersetzen wollen oder Probleme mit einer unserer Applikationen in einer internationalen Umgebung haben, sollten Sie sich in diese Liste eintragen.	
cvs-log@otrs.org	Benachrichtigungen über geänderte CVS-Dateien im Quellcode von OTRS mit sehr viel Verkehr.	

Um eine Liste zu abonnieren besuchen Sie folgenden Link: .

3. Fehler berichten

To submit bugs visit (see Figure below). Please take note of the difference between a bug and a configuration issue. Configuration issues are problems that you encounter when setting a system, or general questions on the use of OTRS. Bug reports should only be used for issues with the source code of OTRS itself, or to file enhancements for OTRS. All your bug reports and enhancement requests are very welcome in the bug tracker.



Für Konfigurationsprobleme sollten Sie entweder den oder die öffentlichen Mailinglisten verwenden.

Wenn Sie uns Fehler berichten, tragen Sie zur Verbesserung des Produkts bei. Wir freuen uns auf Ihre Mitwirkung!

4. Kommerzieller Support

Für Dienstleistungen (Support, Consulting, Entwicklung und Schulungen) können Sie die Firma hinter OTRS kontaktieren, die OTRS AG. Sie hat Niederlassungen in Deutschland, USA, Mexiko, Holland und weiteren Ländern. Schauen Sie sich die Website an, um Kontaktinformationen zu erhalten: .

Anhang B. Referenz der Konfigurationsoptionen

1. Framework

1.1. Core

1.1.1. SecureMode

Description	Value
Description:	Disables the web installer (http://yourhost.example.com/otrs/installer.pl), to prevent the system from being hijacked. If set to "No", the system can be reinstalled and the current basic configuration will be used to pre-populate the questions within the installer script. If not active, it also disables the GenericAgent, PackageManager and SQL Box (to avoid the use of destructive queries, such as DROP DATABASE, and also to steal user passwords).
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SecureMode'} =  '0';`

1.1.2. Frontend::DebugMode

Description	Value
Description:	Enables or disable the debug mode over frontend interface.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::DebugMode'} =  '0';`

1.1.3. ConfigLevel

Description	Value
Description:	Sets the configuration level of the administrator. Depending on the config level, some sysconfig options will be not shown. The config levels are in in ascending order: Expert, Advanced, Beginner. The higher the config level is (e.g. Beginner is the highest), the less likely is it that the user can accidentally configure the system in a way that it is not usable any more.
Group:	Framework
SubGroup:	Core
Valid:	1

Description	Value
Required:	1
Config-Setting:	`$Self->{'ConfigLevel'} = '100';`

1.1.4. ProductName

Description	Value
Description:	Defines the name of the application, shown in the web interface, tabs and title bar of the web browser.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'ProductName'} = 'OTRS';`

1.1.5. SystemID

Description	Value
Description:	Defines the system identifier. Every ticket number and http session string contain this ID. This ensures that only tickets which belong to your system will be processed as follow-ups (useful when communicating between two instances of OTRS).
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SystemID'} = '10';`

1.1.6. FQDN

Description	Value
Description:	Defines the fully qualified domain name of the system. This setting is used as a variable, OTRS_CONFIG_FQDN which is found in all forms of messaging used by the application, to build links to the tickets within your system.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'FQDN'} = 'yourhost.example.com';`

1.1.7. HttpType

Description	Value
Description:	Defines the type of protocol, used by ther web server, to serve the application. If https protocol will be used instead of plain http, it must be specified it here. Since this has no affect on the web server's settings or behavior, it will not change the method of access to the application and, if it is wrong, it will not prevent you from logging into the application. This setting is used as a variable, OTRS_CONFIG_HttpType which is found in all forms of messaging used by the application, to build links to the tickets within your system.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'HttpType'} = 'http';`

1.1.8. ScriptAlias

Description	Value
Description:	Sets the prefix to the scripts folder on the server, as configured on the web server. This setting is used as a variable, OTRS_CONFIG_ScriptAlias which is found in all forms of messaging used by the application, to build links to the tickets within the system.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'ScriptAlias'} = 'otrs/';`

1.1.9. AdminEmail

Description	Value
Description:	Defines the system administrator's email address. It will be displayed in the error screens of the application.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'AdminEmail'} = 'admin@example.com';`

1.1.10. Organization

Description	Value
Description:	Company name for the customer web interface. Will also be included in emails as an X-Header.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Organization'} = 'Example Company';`

1.1.11. DefaultLanguage

Description	Value
Description:	Defines the default front-end language. All the possible values are determined by the available language files on the system (see the next setting).
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'DefaultLanguage'} = 'en';`

1.1.12. DefaultUsedLanguages

Description	Value
Description:	Defines all the languages that are available to the application. The Key/Content pair links the front-end display name to the appropriate language PM file. The "Key" value should be the base-name of the PM file (i.e. de.pm is the file, then de is the "Key" value). The "Content" value should be the display name for the front-end. Specify any own-defined language here (see the developer documentation http://doc.otrs.org/ for more infomation). Please remember to use the HTML equivalents for non-ASCII characters (i.e. for the German oe = o umlaut, it is necessary to use the ö symbol).
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	

Description	Value

	``
	$Self->{'DefaultUsedLanguages'} = {
	  'ar_SA' => 'Arabic (Saudi Arabia)',
	  'bg' => 'Bulgarian
	(Български)',
	  'ca' => 'Català',
	  'cs' => 'Czech (Česky)',
	  'da' => 'Dansk',
	  'de' => 'Deutsch',
	  'el' => 'Greek
	(Ελληνικά)',
	  'en' => 'English (United States)',
	  'en_CA' => 'English (Canada)',
	  'en_GB' => 'English (United Kingdom)',
	  'es' => 'Español',
	  'es_CO' => 'Español (Colombia)',
	  'es_MX' => 'Español (México)',
	  'et' => 'Eesti',
	  'fa' => 'Persian (فارسى)',
	  'fi' => 'Suomi',
	  'fr' => 'Français',
	  'hi' => 'Hindi',
	  'hr' => 'Hrvatski',
	  'hu' => 'Magyar',
	  'it' => 'Italiano',
	  'ja' => 'Japanese (日本&#x8a9e)',
	  'lt' => "Lietuvi\x{173} kalba",
	  'lv' => 'Latvijas',
	  'ms' => 'Malay',
	  'nb_NO' => 'Norsk bokmål',
	  'nl' => 'Nederlands',
	  'pl' => 'Polski',
	  'pt' => 'Português',
	  'pt_BR' => 'Português Brasileiro',
	  'ru' => 'Russian
	(Русский)',
	  'sk_SK' => 'Slovak (Slovenčina)',
	  'sl' => "Slovenian (Sloven\x{161}\x{10d}ina)",
	  'sr_Cyrl' => "Serbian Cyrillic
	(\x{441}\x{440}\x{43f}\x{441}\x{43a}\x{438})",
	  'sr_Latn' => 'Serbian Latin (Srpski)',
	  'sv' => 'Svenska',
	  'tr' => 'Türkçe',
	  'uk' => 'Ukrainian
	(Українська)',
	  'vi_VN' => 'Vietnam (ViɆt Nam)',
	  'zh_CN' => 'Chinese (Sim.) (简体中文)',
	  'zh_TW' => 'Chinese (Tradi.) (正體中文)'
	};
	``

1.1.13. DefaultTheme

Description	Value
Description:	Defines the default front-end (HTML) theme to be used by the agents and customers. The default themes are Standard and Lite. If you like, you can add your own theme. Please refer the administrator manual located at http://doc.otrs.org/.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1

Description	Value
Config-Setting:	`$Self->{'DefaultTheme'} =  'Standard';`

1.1.14. DefaultTheme::HostBased

Description	Value
Description:	It is possible to configure different themes, for example to distinguish between agents and customers, to be used on a per-domain basis within the application. Using a regular expression (regex), you can configure a Key/Content pair to match a domain. The value in "Key" should match the domain, and the value in "Content" should be a valid theme on your system. Please see the example entries for the proper form of the regex.
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0

Config-Setting:	`$Self->{'DefaultTheme::HostBased'} =  {`
	`  'host1\\.example\\.com' => 'SomeTheme1',`
	`  'host2\\.example\\.com' => 'SomeTheme2'`
	`};`

1.1.15. CheckMXRecord

Description	Value
Description:	Makes the application check the MX record of email addresses before sending an email or submitting a telephone or email ticket.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CheckMXRecord'} =  '1';`

1.1.16. CheckMXRecord::Nameserver

Description	Value
Description:	Defines the address of a dedicated DNS server, if necessary, for the "CheckMXRecord" look-ups.
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'CheckMXRecord::Nameserver'} = 'ns.example.com';`

1.1.17. CheckEmailAddresses

Description	Value
Description:	Makes the application check the syntax of email addresses.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CheckEmailAddresses'} = '1';`

1.1.18. CheckEmailValidAddress

Description	Value
Description:	Defines a regular expression that excludes some addresses from the syntax check (if "CheckEmailAddresses" is set to "Yes"). Please enter a regex in this field for email addresses, that aren't syntactically valid, but are necessary for the system (i.e. "root@localhost").
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CheckEmailValidAddress'} = '^(root@localhost\|admin@localhost)$';`

1.1.19. CheckEmailInvalidAddress

Description	Value
Description:	Defines a regular expression that filters all email addresses that should not be used in the application.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CheckEmailInvalidAddress'} = '@(example)\\.(..\|...)$';`

1.1.20. CGILogPrefix

Description	Value
Description:	Specifies the text that should appear in the log file to denote a CGI script entry.

Description	Value
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CGILogPrefix'} = 'OTRS-CGI';`

1.1.21. DemoSystem

Description	Value
Description:	Runs the system in "Demo" mode. If set to "Yes", agents can change preferences, such as selection of language and theme via the agent web interface. These changes are only valid for the current session. It will not be possible for agents to change their passwords.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'DemoSystem'} = '0';`

1.1.22. SwitchToUser

Description	Value
Description:	Allows the administrators to login as other users, via the users administration panel.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SwitchToUser'} = '0';`

1.1.23. NotificationSenderName

Description	Value
Description:	Specifies the name that should be used by the application when sending notifications. The sender name is used to build the complete display name for the notification master (i.e. "OTRS Notification Master" otrs@your.example.com). Notifications are messages such as en::Customer::QueueUpdate or en::Agent::Move.
Group:	Framework
SubGroup:	Core

Description	Value
Valid:	1
Required:	1
Config-Setting:	`$Self->{'NotificationSenderName'} = 'OTRS Notification Master';`

1.1.24. NotificationSenderEmail

Description	Value
Description:	Specifies the email address that should be used by the application when sending notifications. The email address is used to build the complete display name for the notification master (i.e. "OTRS Notification Master" otrs@your.example.com). You can use the OTRS_CONFIG_FQDN variable as set in your configuation, or choose another email address. Notifications are messages such as en::Customer::QueueUpdate or en::Agent::Move.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'NotificationSenderEmail'} = 'otrs@<OTRS_CONFIG_FQDN>';`

1.1.25. System::Customer::Permission

Description	Value
Description:	Defines the standard permissions available for customers within the application. If more permissions are needed, you can enter them here. Permissions must be hard coded to be effective. Please ensure, when adding any of the afore mentioned permissions, that the "rw" permission remains the last entry.
Group:	Framework
SubGroup:	Core
Valid:	1
Required:	1
Config-Setting:	`$Self->{'System::Customer::Permission'} = [ 'ro', 'rw' ];`

1.1.26. LanguageDebug

Description	Value
Description:	Debugs the translation set. If this is set to "Yes" all strings (text) without translations are written to STDERR. This can be helpful when

Description	Value
	you are creating a new translation file. Otherwise, this option should remain set to "No".
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	`$Self->{'LanguageDebug'} = '0';`

1.1.27. Secure::DisableBanner

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

1.1.28. StandardResponse2QueueByCreating

Description	Value
Description:	List of default StandardResponses which are assigned automatically to new Queues upon creation.
Group:	Framework
SubGroup:	Core
Valid:	0
Required:	0
Config-Setting:	`$Self->{'StandardResponse2QueueByCreating'} = [ '' ];`

1.2. Core::LinkObject

1.2.1. LinkObject::ViewMode

Description	Value
Description:	Determines the way the linked objects are displayed in each zoom mask.
Group:	Framework
SubGroup:	Core::LinkObject

Description	Value
Valid:	1
Required:	1
Config-Setting:	`$Self->{'LinkObject::ViewMode'} = 'Simple';`

1.2.2. LinkObject::Type###Normal

Description	Value
Description:	Defines the link type 'Normal'. If the source name and the target name contain the same value, the resulting link is a non-directional one; otherwise, the result is a directional link.
Group:	Framework
SubGroup:	Core::LinkObject
Valid:	1
Required:	1
Config-Setting:	`$Self->{'LinkObject::Type'}->{'Normal'} = { 'SourceName' => 'Normal', 'TargetName' => 'Normal' };`

1.2.3. LinkObject::Type###ParentChild

Description	Value
Description:	Defines the link type 'ParentChild'. If the source name and the target name contain the same value, the resulting link is a non-directional one; otherwise, the result is a directional link.
Group:	Framework
SubGroup:	Core::LinkObject
Valid:	1
Required:	1
Config-Setting:	`$Self->{'LinkObject::Type'}->{'ParentChild'} = { 'SourceName' => 'Parent', 'TargetName' => 'Child' };`

1.2.4. LinkObject::TypeGroup###0001

Description	Value
Description:	Defines the link type groups. The link types of the same group cancel one another. Example: If ticket A is linked per a 'Normal' link with ticket B, then these tickets could not be additionally linked with link of a 'ParentChild' relationship.
Group:	Framework
SubGroup:	Core::LinkObject

Description	Value
Valid:	1
Required:	0
Config-Setting:	`$Self->{'LinkObject::TypeGroup'}->{'0001'} = [ 'Normal', 'ParentChild' ];`

1.3. Core::Log

1.3.1. LogModule

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

1.3.2. LogModule::SysLog::Facility

Description	Value
Description:	If "SysLog" was selected for LogModule, a special log facility can be specified.
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	`$Self->{'LogModule::SysLog::Facility'} = 'user';`

1.3.3. LogModule::SysLog::LogSock

Description	Value
Description:	If "SysLog" was selected for LogModule, a special log sock can be specified (on solaris you may need to use 'stream').
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'LogModule::SysLog::LogSock'} = 'unix';`

1.3.4. LogModule::SysLog::Charset

Description	Value
Description:	If "SysLog" was selected for LogModule, the charset that should be used for logging can be specified.
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	`$Self->{'LogModule::SysLog::Charset'} = 'utf-8';`

1.3.5. LogModule::LogFile

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

1.3.6. LogModule::LogFile::Date

Description	Value
Description:	Adds a suffix with the actual year and month to the OTRS log file. A logfile for every month will be created.
Group:	Framework
SubGroup:	Core::Log
Valid:	1
Required:	1
Config-Setting:	`$Self->{'LogModule::LogFile::Date'} = '0';`

1.4. Core::MIME-Viewer

1.4.1. MIME-Viewer###application/excel

Description	Value
Description:	Specifies the path to the converter that allows the view of Microsoft Excel files, in the web interface.

Description	Value
Group:	Framework
SubGroup:	Core::MIME-Viewer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'MIME-Viewer'}->{'application/excel'} = 'xlhtml';`

1.4.2. MIME-Viewer###application/msword

Description	Value
Description:	Specifies the path to the converter that allows the view of Microsoft Word files, in the web interface.
Group:	Framework
SubGroup:	Core::MIME-Viewer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'MIME-Viewer'}->{'application/msword'} = 'wvWare';`

1.4.3. MIME-Viewer###application/pdf

Description	Value
Description:	Specifies the path to the converter that allows the view of PDF documents, in the web interface.
Group:	Framework
SubGroup:	Core::MIME-Viewer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'MIME-Viewer'}->{'application/pdf'} = 'pdftohtml -stdout -i';`

1.4.4. MIME-Viewer###text/xml

Description	Value
Description:	Specifies the path to the converter that allows the view of XML files, in the web interface.
Group:	Framework
SubGroup:	Core::MIME-Viewer
Valid:	0
Required:	0

Description	Value
Config-Setting:	`$Self->{'MIME-Viewer'}->{'text/xml'} = '<OTRS_CONFIG_Home>/scripts/tools/xml2html.pl';`

1.5. Core::MirrorDB

1.5.1. Core::MirrorDB::DSN

Description	Value
Description:	If you want to use a mirror database for agent ticket fulltext search or to generate stats, specify the DSN to this database.
Group:	Framework
SubGroup:	Core::MirrorDB
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Core::MirrorDB::DSN'} = 'DBI:mysql:database=mirrordb;host=mirrordbhost';`

1.5.2. Core::MirrorDB::User

Description	Value
Description:	If you want to use a mirror database for agent ticket fulltext search or to generate stats, the user to authenticate to this database can be specified.
Group:	Framework
SubGroup:	Core::MirrorDB
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Core::MirrorDB::User'} = 'some_user';`

1.5.3. Core::MirrorDB::Password

Description	Value
Description:	If you want to use a mirror database for agent ticket fulltext search or to generate stats, the password to authenticate to this database can be specified.
Group:	Framework
SubGroup:	Core::MirrorDB
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'Core::MirrorDB::Password'} = 'some_password';`

1.6. Core::PDF

1.6.1. PDF

Description	Value
Description:	Enables PDF output. The CPAN module PDF::API2 is required, if not installed, PDF output will be disabled.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PDF'} = '1';`

1.6.2. PDF::LogoFile

Description	Value
Description:	Specifies the path of the file for the logo in the page header (gif|jpg|png, 700 x 100 pixel).
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PDF::LogoFile'} = '<OTRS_CONFIG_Home>/var/logo-otrs.png';`

1.6.3. PDF::PageSize

Description	Value
Description:	Defines the standard size of PDF pages.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PDF::PageSize'} = 'a4';`

1.6.4. PDF::MaxPages

Description	Value
Description:	Defines the maximum number of pages per PDF file.

Description	Value
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PDF::MaxPages'} = '100';`

1.6.5. PDF::TTFontFile###Proportional

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

1.6.6. PDF::TTFontFile###ProportionalBold

Description	Value
Description:	Defines the path and TTF-File to handle bold proportional font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PDF::TTFontFile'}->{'ProportionalBold'} = 'DejaVuSans-Bold.ttf';`

1.6.7. PDF::TTFontFile###ProportionalItalic

Description	Value
Description:	Defines the path and TTF-File to handle italic proportional font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'PDF::TTFontFile'}->{'ProportionalItalic'} = 'DejaVuSans-Oblique.ttf';`

1.6.8. PDF::TTFontFile###ProportionalBoldItalic

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

1.6.9. PDF::TTFontFile###Monospaced

Description	Value
Description:	Defines the path and TTF-File to handle monospaced font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PDF::TTFontFile'}->{'Monospaced'} = 'DejaVuSansMono.ttf';`

1.6.10. PDF::TTFontFile###MonospacedBold

Description	Value
Description:	Defines the path and TTF-File to handle bold monospaced font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	0
Required:	0
Config-Setting:	`$Self->{'PDF::TTFontFile'}->{'MonospacedBold'} = 'DejaVuSansMono-Bold.ttf';`

1.6.11. PDF::TTFontFile###MonospacedItalic

Description	Value
Description:	Defines the path and TTF-File to handle italic monospaced font in PDF documents.

Description	Value
Group:	Framework
SubGroup:	Core::PDF
Valid:	0
Required:	0
Config-Setting:	`$Self->{'PDF::TTFontFile'}->{'MonospacedItalic'} = 'DejaVuSansMono-Oblique.ttf';`

1.6.12. PDF::TTFontFile###MonospacedBoldItalic

Description	Value
Description:	Defines the path and TTF-File to handle bold italic monospaced font in PDF documents.
Group:	Framework
SubGroup:	Core::PDF
Valid:	0
Required:	0
Config-Setting:	`$Self->{'PDF::TTFontFile'}->{'MonospacedBoldItalic'} = 'DejaVuSansMono-BoldOblique.ttf';`

1.7. Core::Package

1.7.1. Package::FileUpload

Description	Value
Description:	Enables file upload in the package manager frontend.
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Package::FileUpload'} = '1';`

1.7.2. Package::RepositoryRoot

Description	Value
Description:	Defines the location to get online repository list for additional packages. The first available result will be used.
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	0
Config-Setting:	

Description	Value

	`$Self->{'Package::RepositoryRoot'} = [`
	`'http://ftp.otrs.org/pub/otrs/misc/packages/repository.xml',`
	`'http://otrs.org/repository.xml'`
	`];`

1.7.3. Package::RepositoryList

Description	Value
Description:	Defines the list of online repositories. Another installations can be used as repository, for example: Key="http://example.com/otrs/public.pl?Action=PublicRepository;File=" and Content="Some Name".
Group:	Framework
SubGroup:	Core::Package
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Package::RepositoryList'} = {`
	`'ftp://ftp.example.com/pub/otrs/misc/packages/' => '[Example] ftp://ftp.example.com/'`
	`};`

1.7.4. Package::RepositoryAccessRegExp

Description	Value
Description:	Defines the IP regular expression for accessing the local repository. You need to enable this to have access to your local repository and the package::RepositoryList is required on the remote host.
Group:	Framework
SubGroup:	Core::Package
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Package::RepositoryAccessRegExp'} = '127\\.0\\.0\\.1';`

1.7.5. Package::Timeout

Description	Value
Description:	Sets the timeout (in seconds) for package downloads. Overwrites "WebUserAgent::Timeout".
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'Package::Timeout'} = '15';`

1.7.6. Package::Proxy

Description	Value
Description:	Fetches packages via proxy. Overwrites "WebUserAgent::Proxy".
Group:	Framework
SubGroup:	Core::Package
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Package::Proxy'} = 'http://proxy.sn.no:8001/';`

1.7.7. Package::ShowFeatureAddons

Description	Value
Description:	Toggles display of OTRS FeatureAddons list in PackageManager.
Group:	Framework
SubGroup:	Core::Package
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Package::ShowFeatureAddons'} = '1';`

1.8. Core::PerformanceLog

1.8.1. PerformanceLog

Description	Value
Description:	Enables performance log (to log the page response time). It will affect the system performance. Frontend::Module###AdminPerformanceLog must be enabled.
Group:	Framework
SubGroup:	Core::PerformanceLog
Valid:	1
Required:	0
Config-Setting:	`$Self->{'PerformanceLog'} = '0';`

1.8.2. PerformanceLog::File

Description	Value
Description:	Specifies the path of the file for the performance log.

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

1.8.3. PerformanceLog::FileMax

Description	Value
Description:	Defines the maximum size (in MB) of the log file.
Group:	Framework
SubGroup:	Core::PerformanceLog
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PerformanceLog::FileMax'} = '25';`

1.9. Core::ReferenceData

1.9.1. ReferenceData::OwnCountryList

Description	Value
Description:	This setting allows you to override the built-in country list with your own list of countries. This is particularly handy if you just want to use a small select group of countries.
Group:	Framework
SubGroup:	Core::ReferenceData
Valid:	0
Required:	0
Config-Setting:	`$Self->{'ReferenceData::OwnCountryList'} = { 'AT' => 'Austria', 'CH' => 'Switzerland', 'DE' => 'Germany' };`

1.10. Core::SOAP

1.10.1. SOAP::User

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

Description	Value
SubGroup:	Core::SOAP
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SOAP::User'} = 'some_user';`

1.10.2. SOAP::Password

Description	Value
Description:	Defines the password to access the SOAP handle (bin/cgi-bin/rpc.pl).
Group:	Framework
SubGroup:	Core::SOAP
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SOAP::Password'} = 'some_pass';`

1.10.3. SOAP::Keep-Alive

Description	Value
Description:	Enable keep-alive connection header for SOAP responses.
Group:	Framework
SubGroup:	Core::SOAP
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SOAP::Keep-Alive'} = '0';`

1.11. Core::Sendmail

1.11.1. SendmailModule

Description	Value
Description:	Defines the module to send emails. "Sendmail" directly uses the sendmail binary of your operating system. Any of the "SMTP" mechanisms use a specified (external) mailserver. "DoNotSendEmail" doesn't send emails and it is useful for test systems.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'SendmailModule'} = 'Kernel::System::Email::Sendmail';`

1.11.2. SendmailModule::CMD

Description	Value
Description:	If "Sendmail" was selected as SendmailModule, the location of the sendmail binary and the needed options must be specified.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SendmailModule::CMD'} = '/usr/sbin/sendmail -i -f';`

1.11.3. SendmailModule::Host

Description	Value
Description:	If any of the "SMTP" mechanisms was selected as SendmailModule, the mailhost that sends out the mails must be specified.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SendmailModule::Host'} = 'mail.example.com';`

1.11.4. SendmailModule::Port

Description	Value
Description:	If any of the "SMTP" mechanisms was selected as SendmailModule, the port where your mailserver is listening for incoming connections must be specified.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SendmailModule::Port'} = '25';`

1.11.5. SendmailModule::AuthUser

Description	Value
Description:	If any of the "SMTP" mechanisms was selected as SendmailModule, and authentication to the mail server is needed, an username must be specified.

Description	Value
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SendmailModule::AuthUser'} = 'MailserverLogin';`

1.11.6. SendmailModule::AuthPassword

Description	Value
Description:	If any of the "SMTP" mechanisms was selected as SendmailModule, and authentication to the mail server is needed, a password must be specified.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	`$Self->{'SendmailModule::AuthPassword'} = 'MailserverPassword';`

1.11.7. SendmailBcc

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

1.11.8. SendmailNotificationEnvelopeFrom

Description	Value
Description:	If set, this address is used as envelope sender header in outgoing notifications. If no address is specified, the envelope sender header is empty.
Group:	Framework
SubGroup:	Core::Sendmail
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'SendmailNotificationEnvelopeFrom'} = '';`

1.11.9. SendmailEncodingForce

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

1.12. Core::Session

1.12.1. SessionModule

Description	Value
Description:	Defines the module used to store the session data. With "DB" the frontend server can be splitted from the db server. "FS" is faster.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionModule'} = 'Kernel::System::AuthSession::DB';`

1.12.2. SessionName

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

1.12.3. SessionCheckRemoteIP

Description	Value
Description:	Turns on the remote ip address check. It should be set to "No" if the application is used, for example, via a proxy farm or a dialup

Description	Value
	connection, because the remote ip address is mostly different for the requests.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionCheckRemoteIP'} =  '1';`

1.12.4. SessionDeleteIfNotRemoteID

Description	Value
Description:	Deletes a session if the session id is used with an invalid remote IP address.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionDeleteIfNotRemoteID'} =  '1';`

1.12.5. SessionMaxTime

Description	Value
Description:	Defines the maximal valid time (in seconds) for a session id.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionMaxTime'} =  '57600';`

1.12.6. SessionMaxIdleTime

Description	Value
Description:	Sets the inactivity time (in seconds) to pass before a session is killed and a user is loged out.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'SessionMaxIdleTime'} = '21600';`

1.12.7. SessionDeleteIfTimeToOld

Description	Value
Description:	Deletes requested sessions if they have timed out.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionDeleteIfTimeToOld'} = '1';`

1.12.8. SessionUseCookie

Description	Value
Description:	Makes the session management use html cookies. If html cookies are disabled or if the client browser disabled html cookies, then the system will work as usual and append the session id to the links.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionUseCookie'} = '1';`

1.12.9. SessionUseCookieAfterBrowserClose

Description	Value
Description:	Stores cookies after the browser has been closed.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionUseCookieAfterBrowserClose'} = '0';`

1.12.10. SessionCSRFProtection

Description	Value
Description:	Protection against CSRF (Cross Site Request Forgery) exploits (for more info see http://en.wikipedia.org/wiki/Cross-site_request_forgery).

Description	Value
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionCSRFProtection'} = '1';`

1.12.11. SessionDir

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

1.12.12. SessionTable

Description	Value
Description:	If "DB" was selected for SessionModule, a table in database where session data will be stored must be specified.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionTable'} = 'sessions';`

1.12.13. SessionTableID

Description	Value
Description:	If "DB" was selected for SessionModule, a column for the identifiers in session table must be specified.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'SessionTableID'} =  'session_id';`

1.12.14. SessionTableValue

Description	Value
Description:	If "DB" was selected for SessionModule, a column for the values in session table must be specified.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionTableValue'} =  'session_value';`

1.12.15. SessionSaveLoginPasswd

Description	Value
Description:	Saves the login and password on the session table in the database, if "DB" was selected for SessionModule.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SessionSaveLoginPasswd'} =  '0';`

1.12.16. CustomerPanelSessionName

Description	Value
Description:	Defines the name of the key for customer sessions.
Group:	Framework
SubGroup:	Core::Session
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelSessionName'} =  'CSID';`

1.13. Core::SpellChecker

1.13.1. SpellChecker

Description	Value
Description:	Enables spell checker support.

Description	Value
Group:	Framework
SubGroup:	Core::SpellChecker
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SpellChecker'} = '0';`

1.13.2. SpellCheckerBin

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

1.13.3. SpellCheckerDictDefault

Description	Value
Description:	Defines the default spell checker dictionary.
Group:	Framework
SubGroup:	Core::SpellChecker
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SpellCheckerDictDefault'} = 'english';`

1.13.4. SpellCheckerIgnore

Description	Value
Description:	Defines a default list of words, that are ignored by the spell checker.
Group:	Framework
SubGroup:	Core::SpellChecker
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SpellCheckerIgnore'} = [`

Description	Value

	`'www',`
	`'webmail',`
	`'https',`
	`'http',`
	`'html',`
	`'rfc'`
	`];`

1.14. Core::Stats

1.14.1. Stats::StatsHook

Description	Value
Description:	Sets the stats hook.
Group:	Framework
SubGroup:	Core::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::StatsHook'} = 'Stat#';`

1.14.2. Stats::StatsStartNumber

Description	Value
Description:	Start number for statistics counting. Every new stat increments this number.
Group:	Framework
SubGroup:	Core::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::StatsStartNumber'} = '10000';`

1.14.3. Stats::MaxXaxisAttributes

Description	Value
Description:	Defines the default maximum number of X-axis attributes for the time scale.
Group:	Framework
SubGroup:	Core::Stats
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Stats::MaxXaxisAttributes'} = '1000';`

1.15. Core::Stats::Graph

1.15.1. Stats::Graph::t_margin

Description	Value
Description:	Specifies the top margin of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::t_margin'} =  '10';`

1.15.2. Stats::Graph::l_margin

Description	Value
Description:	Specifies the left margin of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::l_margin'} =  '10';`

1.15.3. Stats::Graph::b_margin

Description	Value
Description:	Specifies the bottom margin of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::b_margin'} =  '10';`

1.15.4. Stats::Graph::r_margin

Description	Value
Description:	Specifies the right margin of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1

Description	Value
Config-Setting:	`$Self->{'Stats::Graph::r_margin'} = '20';`

1.15.5. Stats::Graph::bgclr

Description	Value
Description:	Specifies the background color of the picture.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::bgclr'} = 'white';`

1.15.6. Stats::Graph::transparent

Description	Value
Description:	Makes the picture transparent.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::transparent'} = '0';`

1.15.7. Stats::Graph::fgclr

Description	Value
Description:	Specifies the border color of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::fgclr'} = 'black';`

1.15.8. Stats::Graph::boxclr

Description	Value
Description:	Specifies the background color of the chart.
Group:	Framework
SubGroup:	Core::Stats::Graph

Description	Value
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::boxclr'} = 'white';`

1.15.9. Stats::Graph::accentclr

Description	Value
Description:	Specifies the border color of the legend.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::accentclr'} = 'black';`

1.15.10. Stats::Graph::legendclr

Description	Value
Description:	Specifies the text color of the legend.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::legendclr'} = 'black';`

1.15.11. Stats::Graph::textclr

Description	Value
Description:	Specifies the text color of the chart (e. g. caption).
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::textclr'} = 'black';`

1.15.12. Stats::Graph::dclrs

Description	Value
Description:	Defines the colors for the graphs.

Description	Value
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::dclrs'} = [ 'red', 'green', 'blue', 'yellow', 'black', 'purple', 'orange', 'pink', 'marine', 'cyan', 'lgray', 'lblue', 'lyellow', 'lgreen', 'lred', 'lpurple', 'lorange', 'lbrown' ];`

1.15.13. Stats::Graph::line_width

Description	Value
Description:	Defines the boldness of the line drawed by the graph.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::line_width'} = '1';`

1.15.14. Stats::Graph::legend_placement

Description	Value
Description:	Defines the placement of the legend. This should be a two letter key of the form: 'B[LCR]|R[TCB]'. The first letter indicates the placement (Bottom or Right), and the second letter the alignment (Left, Right, Center, Top, or Bottom).
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'Stats::Graph::legend_placement'} = 'BC';`

1.15.15. Stats::Graph::legend_spacing

Description	Value
Description:	Defines the spacing of the legends.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::legend_spacing'} = '4';`

1.15.16. Stats::Graph::legend_marker_width

Description	Value
Description:	Defines the width of the legend.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::legend_marker_width'} = '12';`

1.15.17. Stats::Graph::legend_marker_height

Description	Value
Description:	Defines the height of the legend.
Group:	Framework
SubGroup:	Core::Stats::Graph
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::Graph::legend_marker_height'} = '8';`

1.16. Core::Time

1.16.1. TimeInputFormat

Description	Value
Description:	Defines the date input format used in forms (option or input fields).

Description	Value
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeInputFormat'} = 'Option';`

1.16.2. TimeShowAlwaysLong

Description	Value
Description:	Shows time in long format (days, hours, minutes), if set to "Yes"; or in short format (days, hours), if set to "No".
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeShowAlwaysLong'} = '0';`

1.16.3. TimeZone

Description	Value
Description:	Sets the system time zone (required a system with UTC as system time). Otherwise this is a diff time to the local time.
Group:	Framework
SubGroup:	Core::Time
Valid:	0
Required:	0
Config-Setting:	`$Self->{'TimeZone'} = '+0';`

1.16.4. TimeZoneUser

Description	Value
Description:	Sets the user time zone per user (required a system with UTC as system time and UTC under TimeZone). Otherwise this is a diff time to the local time.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'TimeZoneUser'} = '0';`

1.16.5. TimeZoneUserBrowserAutoOffset

Description	Value
Description:	Sets the user time zone per user based on java script / browser time zone offset feature at login time.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	0
Config-Setting:	`$Self->{'TimeZoneUserBrowserAutoOffset'} = '1';`

1.16.6. CalendarWeekDayStart

Description	Value
Description:	Define the start day of the week for the date picker.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CalendarWeekDayStart'} = '1';`

1.16.7. TimeVacationDays

Description	Value
Description:	Adds the permanent vacation days. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'TimeVacationDays'} = {
 '1' => {
   '1' => 'New Year\'s Day'
 },
 '12' => {
   '24' => 'Christmas Eve',
   '25' => 'First Christmas Day',
   '26' => 'Second Christmas Day',
   '31' => 'New Year\'s Eve'
 },
 '5' => {
   '1' => 'International Workers\' Day'
 }
};
```

1.16.8. TimeVacationDaysOneTime

Description	Value
Description:	Adds the one time vacation days. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeVacationDaysOneTime'} = { '2004' => { '1' => { '1' => 'test' } } };`

1.16.9. TimeWorkingHours

Description	Value
Description:	Defines the hours and week days to count the working time.
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeWorkingHours'} = { 'Fri' => [ '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20' ], 'Mon' => [ '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'`

Description	Value

	``
	  ],
	  'Sat' => [],
	  'Sun' => [],
	  'Thu' => [
	  '8',
	  '9',
	  '10',
	  '11',
	  '12',
	  '13',
	  '14',
	  '15',
	  '16',
	  '17',
	  '18',
	  '19',
	  '20'
	  ],
	  'Tue' => [
	  '8',
	  '9',
	  '10',
	  '11',
	  '12',
	  '13',
	  '14',
	  '15',
	  '16',
	  '17',
	  '18',
	  '19',
	  '20'
	  ],
	  'Wed' => [
	  '8',
	  '9',
	  '10',
	  '11',
	  '12',
	  '13',
	  '14',
	  '15',
	  '16',
	  '17',
	  '18',
	  '19',
	  '20'
	  ]
	};
	``

1.16.10. TimeShowCompleteDescription

Description	Value
Description:	Shows time use complete description (days, hours, minutes), if set to "Yes"; or just first letter (d, h, m), if set to "No".
Group:	Framework
SubGroup:	Core::Time
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'TimeShowCompleteDescription'} = '0';`

1.17. Core::Time::Calendar1

1.17.1. TimeZone::Calendar1Name

Description	Value
Description:	Defines the name of the indicated calendar.
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar1Name'} = 'Calendar Name 1';`

1.17.2. TimeZone::Calendar1

Description	Value
Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar1'} = '0';`

1.17.3. TimeVacationDays::Calendar1

Description	Value
Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'TimeVacationDays::Calendar1'} = {
 '1' => {
   '1' => 'New Year\'s Day'
 },
 '12' => {
   '24' => 'Christmas Eve',
   '25' => 'First Christmas Day',
   '26' => 'Second Christmas Day',
   '31' => 'New Year\'s Eve'
 },
 '5' => {
```

Description	Value

	`    '1' => 'International Workers\' Day'`
	`  }`
	`};`

1.17.4. TimeVacationDaysOneTime::Calendar1

Description	Value
Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1

Config-Setting:	`$Self->{'TimeVacationDaysOneTime::Calendar1'} =  {`
	`  '2004' => {`
	`    '1' => {`
	`      '1' => 'test'`
	`    }`
	`  }`
	`};`

1.17.5. TimeWorkingHours::Calendar1

Description	Value
Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar1
Valid:	1
Required:	1

Config-Setting:	`$Self->{'TimeWorkingHours::Calendar1'} =  {`
	`  'Fri' => [`
	`    '8',`
	`    '9',`
	`    '10',`
	`    '11',`
	`    '12',`
	`    '13',`
	`    '14',`
	`    '15',`
	`    '16',`
	`    '17',`
	`    '18',`
	`    '19',`
	`    '20'`
	`  ],`
	`  'Mon' => [`
	`    '8',`
	`    '9',`
	`    '10',`
	`    '11',`
	`    '12',`

Description	Value

	`'13',`
	`'14',`
	`'15',`
	`'16',`
	`'17',`
	`'18',`
	`'19',`
	`'20'`
	`],`
	`'Sat' => [],`
	`'Sun' => [],`
	`'Thu' => [`
	`'8',`
	`'9',`
	`'10',`
	`'11',`
	`'12',`
	`'13',`
	`'14',`
	`'15',`
	`'16',`
	`'17',`
	`'18',`
	`'19',`
	`'20'`
	`],`
	`'Tue' => [`
	`'8',`
	`'9',`
	`'10',`
	`'11',`
	`'12',`
	`'13',`
	`'14',`
	`'15',`
	`'16',`
	`'17',`
	`'18',`
	`'19',`
	`'20'`
	`],`
	`'Wed' => [`
	`'8',`
	`'9',`
	`'10',`
	`'11',`
	`'12',`
	`'13',`
	`'14',`
	`'15',`
	`'16',`
	`'17',`
	`'18',`
	`'19',`
	`'20'`
	`]`
	`};`

1.18. Core::Time::Calendar2

1.18.1. TimeZone::Calendar2Name

Description	Value
Description:	Defines the name of the indicated calendar.

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

1.18.2. TimeZone::Calendar2

Description	Value
Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar2'} = '0';`

1.18.3. TimeVacationDays::Calendar2

Description	Value
Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'TimeVacationDays::Calendar2'} = {
 '1' => {
   '1' => 'New Year\'s Day'
 },
 '12' => {
   '24' => 'Christmas Eve',
   '25' => 'First Christmas Day',
   '26' => 'Second Christmas Day',
   '31' => 'New Year\'s Eve'
 },
 '5' => {
   '1' => 'International Workers\' Day'
 }
};
```

1.18.4. TimeVacationDaysOneTime::Calendar2

Description	Value
Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).

Description	Value
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeVacationDaysOneTime::Calendar2'} = { '2004' => { '1' => { '1' => 'test' } } };`

1.18.5. TimeWorkingHours::Calendar2

Description	Value
Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar2
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeWorkingHours::Calendar2'} = { 'Fri' => [ '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20' ], 'Mon' => [ '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20' ], 'Sat' => [], 'Sun' => [],`

Description	Value

	``
	  'Thu' => [
	    '8',
	    '9',
	    '10',
	    '11',
	    '12',
	    '13',
	    '14',
	    '15',
	    '16',
	    '17',
	    '18',
	    '19',
	    '20'
	  ],
	  'Tue' => [
	    '8',
	    '9',
	    '10',
	    '11',
	    '12',
	    '13',
	    '14',
	    '15',
	    '16',
	    '17',
	    '18',
	    '19',
	    '20'
	  ],
	  'Wed' => [
	    '8',
	    '9',
	    '10',
	    '11',
	    '12',
	    '13',
	    '14',
	    '15',
	    '16',
	    '17',
	    '18',
	    '19',
	    '20'
	  ]
	};
	``

1.19. Core::Time::Calendar3

1.19.1. TimeZone::Calendar3Name

Description	Value
Description:	Defines the name of the indicated calendar.
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1
Config-Setting:	``` $Self->{'TimeZone::Calendar3Name'} = 'Calendar Name 3'; ```

1.19.2. TimeZone::Calendar3

Description	Value
Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar3'} =  '0';`

1.19.3. TimeVacationDays::Calendar3

Description	Value
Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1

Config-Setting:	`$Self->{'TimeVacationDays::Calendar3'} =  {`
	`  '1' => {`
	`    '1' => 'New Year\'s Day'`
	`  },`
	`  '12' => {`
	`    '24' => 'Christmas Eve',`
	`    '25' => 'First Christmas Day',`
	`    '26' => 'Second Christmas Day',`
	`    '31' => 'New Year\'s Eve'`
	`  },`
	`  '5' => {`
	`    '1' => 'International Workers\' Day'`
	`  }`
	`};`

1.19.4. TimeVacationDaysOneTime::Calendar3

Description	Value
Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeVacationDaysOneTime::Calendar3'} =  {`

Description	Value

	`  '2004' => {`
	`    '1' => {`
	`      '1' => 'test'`
	`    }`
	`  }`
	`};`

1.19.5. TimeWorkingHours::Calendar3

Description	Value
Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar3
Valid:	1
Required:	1

Config-Setting:	`$Self->{'TimeWorkingHours::Calendar3'} =  {`
	`  'Fri' => [`
	`    '8',`
	`    '9',`
	`    '10',`
	`    '11',`
	`    '12',`
	`    '13',`
	`    '14',`
	`    '15',`
	`    '16',`
	`    '17',`
	`    '18',`
	`    '19',`
	`    '20'`
	`  ],`
	`  'Mon' => [`
	`    '8',`
	`    '9',`
	`    '10',`
	`    '11',`
	`    '12',`
	`    '13',`
	`    '14',`
	`    '15',`
	`    '16',`
	`    '17',`
	`    '18',`
	`    '19',`
	`    '20'`
	`  ],`
	`  'Sat' => [],`
	`  'Sun' => [],`
	`  'Thu' => [`
	`    '8',`
	`    '9',`
	`    '10',`
	`    '11',`
	`    '12',`
	`    '13',`
	`    '14',`
	`    '15',`

Description	Value

	`'16',`
	`'17',`
	`'18',`
	`'19',`
	`'20'`
	`],`
	`'Tue' => [`
	`'8',`
	`'9',`
	`'10',`
	`'11',`
	`'12',`
	`'13',`
	`'14',`
	`'15',`
	`'16',`
	`'17',`
	`'18',`
	`'19',`
	`'20'`
	`],`
	`'Wed' => [`
	`'8',`
	`'9',`
	`'10',`
	`'11',`
	`'12',`
	`'13',`
	`'14',`
	`'15',`
	`'16',`
	`'17',`
	`'18',`
	`'19',`
	`'20'`
	`]`
	`};`

1.20. Core::Time::Calendar4

1.20.1. TimeZone::Calendar4Name

Description	Value
Description:	Defines the name of the indicated calendar.
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar4Name'} =  'Calendar Name 4';`

1.20.2. TimeZone::Calendar4

Description	Value
Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.

Description	Value
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar4'} =  '0';`

1.20.3. TimeVacationDays::Calendar4

Description	Value
Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1

Config-Setting:	`$Self->{'TimeVacationDays::Calendar4'} =  {`
	`  '1' => {`
	`    '1' => 'New Year\'s Day'`
	`  },`
	`  '12' => {`
	`    '24' => 'Christmas Eve',`
	`    '25' => 'First Christmas Day',`
	`    '26' => 'Second Christmas Day',`
	`    '31' => 'New Year\'s Eve'`
	`  },`
	`  '5' => {`
	`    '1' => 'International Workers\' Day'`
	`  }`
	`};`

1.20.4. TimeVacationDaysOneTime::Calendar4

Description	Value
Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1

Config-Setting:	`$Self->{'TimeVacationDaysOneTime::Calendar4'} =  {`
	`  '2004' => {`
	`    '1' => {`
	`      '1' => 'test'`
	`    }`
	`  }`
	`};`

1.20.5. TimeWorkingHours::Calendar4

Description	Value
Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar4
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'TimeWorkingHours::Calendar4'} =  {
 'Fri' => [
   '8',
   '9',
   '10',
   '11',
   '12',
   '13',
   '14',
   '15',
   '16',
   '17',
   '18',
   '19',
   '20'
 ],
 'Mon' => [
   '8',
   '9',
   '10',
   '11',
   '12',
   '13',
   '14',
   '15',
   '16',
   '17',
   '18',
   '19',
   '20'
 ],
 'Sat' => [],
 'Sun' => [],
 'Thu' => [
   '8',
   '9',
   '10',
   '11',
   '12',
   '13',
   '14',
   '15',
   '16',
   '17',
   '18',
   '19',
   '20'
 ],
 'Tue' => [
   '8',
   '9',
   '10',
```

Description	Value

	`'11',`
	`'12',`
	`'13',`
	`'14',`
	`'15',`
	`'16',`
	`'17',`
	`'18',`
	`'19',`
	`'20'`
	`],`
	`'Wed' => [`
	`'8',`
	`'9',`
	`'10',`
	`'11',`
	`'12',`
	`'13',`
	`'14',`
	`'15',`
	`'16',`
	`'17',`
	`'18',`
	`'19',`
	`'20'`
	`]`
	`};`

1.21. Core::Time::Calendar5

1.21.1. TimeZone::Calendar5Name

Description	Value
Description:	Defines the name of the indicated calendar.
Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar5Name'} =  'Calendar Name 5';`

1.21.2. TimeZone::Calendar5

Description	Value
Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar5'} =  '0';`

1.21.3. TimeVacationDays::Calendar5

Description	Value
Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'TimeVacationDays::Calendar5'} =  {
 '1' => {
   '1' => 'New Year\'s Day'
 },
 '12' => {
   '24' => 'Christmas Eve',
   '25' => 'First Christmas Day',
   '26' => 'Second Christmas Day',
   '31' => 'New Year\'s Eve'
 },
 '5' => {
   '1' => 'International Workers\' Day'
 }
};
```

1.21.4. TimeVacationDaysOneTime::Calendar5

Description	Value
Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'TimeVacationDaysOneTime::Calendar5'} =  {
 '2004' => {
   '1' => {
     '1' => 'test'
   }
 }
};
```

1.21.5. TimeWorkingHours::Calendar5

Description	Value
Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar5
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'TimeWorkingHours::Calendar5'} =  {
```

Description	Value
	'Fri' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Mon' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Sat' => [], 'Sun' => [], 'Thu' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Tue' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Wed' => ['8', '9',

Description	Value

	`'10',`
	`'11',`
	`'12',`
	`'13',`
	`'14',`
	`'15',`
	`'16',`
	`'17',`
	`'18',`
	`'19',`
	`'20'`
	`]`
	`};`

1.22. Core::Time::Calendar6

1.22.1. TimeZone::Calendar6Name

Description	Value
Description:	Defines the name of the indicated calendar.
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar6Name'} = 'Calendar Name 6';`

1.22.2. TimeZone::Calendar6

Description	Value
Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar6'} = '0';`

1.22.3. TimeVacationDays::Calendar6

Description	Value
Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1

Config-Setting:	`$Self->{'TimeVacationDays::Calendar6'} = {`
	`'1' => {`
	`'1' => 'New Year\'s Day'`

Description	Value

	`},`
	`'12' => {`
	`'24' => 'Christmas Eve',`
	`'25' => 'First Christmas Day',`
	`'26' => 'Second Christmas Day',`
	`'31' => 'New Year\'s Eve'`
	`},`
	`'5' => {`
	`'1' => 'International Workers\' Day'`
	`}`
	`};`

1.22.4. TimeVacationDaysOneTime::Calendar6

Description	Value
Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1

Config-Setting:	`$Self->{'TimeVacationDaysOneTime::Calendar6'} = {`
	`'2004' => {`
	`'1' => {`
	`'1' => 'test'`
	`}`
	`}`
	`};`

1.22.5. TimeWorkingHours::Calendar6

Description	Value
Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar6
Valid:	1
Required:	1

Config-Setting:	`$Self->{'TimeWorkingHours::Calendar6'} = {`
	`'Fri' => [`
	`'8',`
	`'9',`
	`'10',`
	`'11',`
	`'12',`
	`'13',`
	`'14',`
	`'15',`
	`'16',`
	`'17',`
	`'18',`
	`'19',`
	`'20'`
	`],`

Description	Value
	`'Mon' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Sat' => [], 'Sun' => [], 'Thu' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Tue' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'], 'Wed' => ['8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20'] };`

1.23. Core::Time::Calendar7

1.23.1. TimeZone::Calendar7Name

Description	Value
Description:	Defines the name of the indicated calendar.
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar7Name'} =  'Calendar Name 7';`

1.23.2. TimeZone::Calendar7

Description	Value
Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar7'} =  '0';`

1.23.3. TimeVacationDays::Calendar7

Description	Value
Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'TimeVacationDays::Calendar7'} =  {
 '1' => {
   '1' => 'New Year\'s Day'
 },
 '12' => {
   '24' => 'Christmas Eve',
   '25' => 'First Christmas Day',
   '26' => 'Second Christmas Day',
   '31' => 'New Year\'s Eve'
 },
 '5' => {
   '1' => 'International Workers\' Day'
 }
};
```

1.23.4. TimeVacationDaysOneTime::Calendar7

Description	Value
Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeVacationDaysOneTime::Calendar7'} = { '2004' => { '1' => { '1' => 'test' } } };`

1.23.5. TimeWorkingHours::Calendar7

Description	Value
Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar7
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeWorkingHours::Calendar7'} = { 'Fri' => [ '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20' ], 'Mon' => [ '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19',`

Description	Value

	`    '20'`
	`  ],`
	`  'Sat' => [],`
	`  'Sun' => [],`
	`  'Thu' => [`
	`    '8',`
	`    '9',`
	`    '10',`
	`    '11',`
	`    '12',`
	`    '13',`
	`    '14',`
	`    '15',`
	`    '16',`
	`    '17',`
	`    '18',`
	`    '19',`
	`    '20'`
	`  ],`
	`  'Tue' => [`
	`    '8',`
	`    '9',`
	`    '10',`
	`    '11',`
	`    '12',`
	`    '13',`
	`    '14',`
	`    '15',`
	`    '16',`
	`    '17',`
	`    '18',`
	`    '19',`
	`    '20'`
	`  ],`
	`  'Wed' => [`
	`    '8',`
	`    '9',`
	`    '10',`
	`    '11',`
	`    '12',`
	`    '13',`
	`    '14',`
	`    '15',`
	`    '16',`
	`    '17',`
	`    '18',`
	`    '19',`
	`    '20'`
	`  ]`
	`};`

1.24. Core::Time::Calendar8

1.24.1. TimeZone::Calendar8Name

Description	Value
Description:	Defines the name of the indicated calendar.
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1

Description	Value
Config-Setting:	`$Self->{'TimeZone::Calendar8Name'} = 'Calendar Name 8';`

1.24.2. TimeZone::Calendar8

Description	Value
Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar8'} = '0';`

1.24.3. TimeVacationDays::Calendar8

Description	Value
Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeVacationDays::Calendar8'} = { '1' => { '1' => 'New Year\'s Day' }, '12' => { '24' => 'Christmas Eve', '25' => 'First Christmas Day', '26' => 'Second Christmas Day', '31' => 'New Year\'s Eve' }, '5' => { '1' => 'International Workers\' Day' } };`

1.24.4. TimeVacationDaysOneTime::Calendar8

Description	Value
Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1

Description	Value
Config-Setting:	`$Self->{'TimeVacationDaysOneTime::Calendar8'} = { '2004' => { '1' => { '1' => 'test' } } };`

1.24.5. TimeWorkingHours::Calendar8

Description	Value
Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar8
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeWorkingHours::Calendar8'} = { 'Fri' => [ '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20' ], 'Mon' => [ '8', '9', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '20' ], 'Sat' => [], 'Sun' => [], 'Thu' => [ '8', '9', '10', '11', '12', '13', '14',`

Description	Value

	``
	    '15',
	    '16',
	    '17',
	    '18',
	    '19',
	    '20'
	  ],
	  'Tue' => [
	    '8',
	    '9',
	    '10',
	    '11',
	    '12',
	    '13',
	    '14',
	    '15',
	    '16',
	    '17',
	    '18',
	    '19',
	    '20'
	  ],
	  'Wed' => [
	    '8',
	    '9',
	    '10',
	    '11',
	    '12',
	    '13',
	    '14',
	    '15',
	    '16',
	    '17',
	    '18',
	    '19',
	    '20'
	  ]
	};
	``

1.25. Core::Time::Calendar9

1.25.1. TimeZone::Calendar9Name

Description	Value
Description:	Defines the name of the indicated calendar.
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar9Name'} =  'Calendar Name 9';`

1.25.2. TimeZone::Calendar9

Description	Value
Description:	Defines the time zone of the indicated calendar, which can be assigned later to a specific queue.

Description	Value
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeZone::Calendar9'} =  '0';`

1.25.3. TimeVacationDays::Calendar9

Description	Value
Description:	Adds the permanent vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeVacationDays::Calendar9'} =  { '1' => { '1' => 'New Year\'s Day' }, '12' => { '24' => 'Christmas Eve', '25' => 'First Christmas Day', '26' => 'Second Christmas Day', '31' => 'New Year\'s Eve' }, '5' => { '1' => 'International Workers\' Day' } };`

1.25.4. TimeVacationDaysOneTime::Calendar9

Description	Value
Description:	Adds the one time vacation days for the indicated calendar. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TimeVacationDaysOneTime::Calendar9'} =  { '2004' => { '1' => { '1' => 'test' } } };`

1.25.5. TimeWorkingHours::Calendar9

Description	Value
Description:	Defines the hours and week days of the indicated calendar, to count the working time.
Group:	Framework
SubGroup:	Core::Time::Calendar9
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'TimeWorkingHours::Calendar9'} =  {
 'Fri' => [
   '8',
   '9',
   '10',
   '11',
   '12',
   '13',
   '14',
   '15',
   '16',
   '17',
   '18',
   '19',
   '20'
 ],
 'Mon' => [
   '8',
   '9',
   '10',
   '11',
   '12',
   '13',
   '14',
   '15',
   '16',
   '17',
   '18',
   '19',
   '20'
 ],
 'Sat' => [],
 'Sun' => [],
 'Thu' => [
   '8',
   '9',
   '10',
   '11',
   '12',
   '13',
   '14',
   '15',
   '16',
   '17',
   '18',
   '19',
   '20'
 ],
 'Tue' => [
   '8',
   '9',
   '10',
```

Description	Value

	`'11',`
	`'12',`
	`'13',`
	`'14',`
	`'15',`
	`'16',`
	`'17',`
	`'18',`
	`'19',`
	`'20'`
	`],`
	`'Wed' => [`
	`'8',`
	`'9',`
	`'10',`
	`'11',`
	`'12',`
	`'13',`
	`'14',`
	`'15',`
	`'16',`
	`'17',`
	`'18',`
	`'19',`
	`'20'`
	`]`
	`};`

1.26. Core::Web

1.26.1. Frontend::WebPath

Description	Value
Description:	Defines the URL base path of icons, CSS and Java Script.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::WebPath'} =  '/otrs-web/';`

1.26.2. Frontend::ImagePath

Description	Value
Description:	Defines the URL image path of icons for navigation.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::ImagePath'} =` `'<OTRS_CONFIG_Frontend::WebPath>skins/Agent/default/img/';`

1.26.3. Frontend::CSSPath

Description	Value
Description:	Defines the URL CSS path.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::CSSPath'} = '<OTRS_CONFIG_Frontend::WebPath>css/';`

1.26.4. Frontend::JavaScriptPath

Description	Value
Description:	Defines the URL java script path.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::JavaScriptPath'} = '<OTRS_CONFIG_Frontend::WebPath>js/';`

1.26.5. Frontend::RichText

Description	Value
Description:	Uses richtext for viewing and editing: articles, salutations, signatures, standard responses, auto responses and notifications.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::RichText'} =  '1';`

1.26.6. Frontend::RichTextPath

Description	Value
Description:	Defines the URL rich text editor path.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::RichTextPath'} = '<OTRS_CONFIG_Frontend::WebPath>js/thirdparty/ckeditor-3.6.6/';`

1.26.7. Frontend::RichTextWidth

Description	Value
Description:	Defines the width for the rich text editor component. Enter number (pixels) or percent value (relative).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::RichTextWidth'} = '620';`

1.26.8. Frontend::RichTextHeight

Description	Value
Description:	Defines the height for the rich text editor component. Enter number (pixels) or percent value (relative).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::RichTextHeight'} = '320';`

1.26.9. Frontend::RichText::DefaultCSS

Description	Value
Description:	Defines the default CSS used in rich text editors.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::RichText::DefaultCSS'} = 'font-family:Geneva,Helvetica,Arial,sans-serif; font-size: 12px;';`

1.26.10. DefaultViewNewLine

Description	Value
Description:	Automated line break in text messages after x number of chars.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'DefaultViewNewLine'} = '90';`

1.26.11. DefaultViewLines

Description	Value
Description:	Sets the number of lines that are displayed in text messages (e.g. ticket lines in the QueueZoom).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'DefaultViewLines'} = '6000';`

1.26.12. Frontend::AnimationEnabled

Description	Value
Description:	Turns on the animations used in the GUI. If you have problems with these animations (e.g. performance issues), you can turn them off here.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::AnimationEnabled'} = '1';`

1.26.13. AttachmentDownloadType

Description	Value
Description:	Allows choosing between showing the attachments of a ticket in the browser (inline) or just make them downloadable (attachment).
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'AttachmentDownloadType'} = 'attachment';`

1.26.14. WebMaxFileUpload

Description	Value
Description:	Defines the maximal size (in bytes) for file uploads via the browser.
Group:	Framework
SubGroup:	Core::Web

Description	Value
Valid:	1
Required:	1
Config-Setting:	`$Self->{'WebMaxFileUpload'} =  '16777216';`

1.26.15. WebUploadCacheModule

Description	Value
Description:	Selects the module to handle uploads via the web interface. "DB" stores all uploads in the database, "FS" uses the file system.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'WebUploadCacheModule'} = 'Kernel::System::Web::UploadCache::DB';`

1.26.16. Frontend::Output::FilterText###AAAURL

Description	Value
Description:	Defines the filter that processes the text in the articles, in order to highlight URLs.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Output::FilterText'}->{'AAAURL'} =  { 'Module' => 'Kernel::Output::HTML::OutputFilterTextURL', 'Templates' => { 'AgentTicketZoom' => '1' } };`

1.26.17. Frontend::Themes

Description	Value
Description:	Activates the available themes on the system. Value 1 means active, 0 means inactive.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Themes'} =  { 'Lite' => '0', 'Standard' => '1'`

Description	Value
	`};`

1.26.18. Frontend::Output::FilterText###OutputFilterTextAutoLink

Description	Value
Description:	Defines a filter to process the text in the articles, in order to highlight predefined keywords.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::Output::FilterText'}->{'OutputFilterTextAutoLink'} = { 'Module' => 'Kernel::Output::HTML::OutputFilterTextAutoLink', 'Templates' => { 'AgentTicketZoom' => '1' } };`

1.26.19. Frontend::Output::OutputFilterTextAutoLink###CVE

Description	Value
Description:	Defines a filter for html output to add links behind CVE numbers. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possiblity is to insert the link to the image.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::Output::OutputFilterTextAutoLink'}->{'CVE'} = { 'RegExp' => ['(CVE|CAN)\\-(\\d{3,4})\\-(\\d{2,6})'], 'Templates' => { 'AgentTicketZoom' => '1' }, 'URL1' => { 'Description' => 'Mitre', 'Image' => 'http://cve.mitre.org/favicon.ico', 'Target' => '_blank', 'URL' => 'http://cve.mitre.org/cgi-bin/cvename.cgi?name=<MATCH1>-<MATCH2>-<MATCH3>' }, 'URL2' => { 'Description' => 'Google', 'Image' => 'http://www.google.de/favicon.ico', 'Target' => '_blank', 'URL' => 'http://google.com/search?q=<MATCH1>-<MATCH2>-<MATCH3>' }, 'URL3' => { 'Description' => 'US-CERT NVD',`

Description	Value

	`'Image' => 'http://nvd.nist.gov/favicon.ico',`
	`'Target' => '_blank',`
	`'URL' => 'http://nvd.nist.gov/nvd.cfm?cvename=<MATCH1>-<MATCH2>-<MATCH3>'`
	`}`
	`};`

1.26.20. Frontend::Output::OutputFilterTextAutoLink###Bugtraq

Description	Value
Description:	Defines a filter for html output to add links behind bugtraq numbers. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possiblity is to insert the link to the image.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::Output::OutputFilterTextAutoLink'}->{'Bugtraq'} =`
	`{`
	`'RegExp' => [`
	`'Bugtraq[\\s\\w\\t]*?ID[\\s\\w\\t]*?:[\\s\\w\\t]*?(\\d{2,8})',`
	`'Bugtraq[\\s\\w\\t]*?ID[\\s\\w\\t]*?(\\d{2,8})',`
	`'Bugtraq[\\s\\w\\t]*?:[\\s\\w\\t]*?(\\d{2,8})',`
	`'Bugtraq[\\s\\w\\t]*?(\\d{2,8})',`
	`'BID[\\s\\w\\t]*?:[\\s\\w\\t]*?(\\d{2,8})',`
	`'BID[\\s\\w\\t]*?(\\d{2,8})'`
	`],`
	`'Templates' => {`
	`'AgentTicketZoom' => '1'`
	`},`
	`'URL1' => {`
	`'Description' => 'Security Focus',`
	`'Image' => 'http://www.securityfocus.com/favicon.ico',`
	`'Target' => '_blank',`
	`'URL' => 'http://www.securityfocus.com/bid/<MATCH1>/info'`
	`},`
	`'URL2' => {`
	`'Description' => 'Google',`
	`'Image' => 'http://www.google.de/favicon.ico',`
	`'Target' => '_blank',`
	`'URL' => 'http://google.com/search?q=<MATCH>'`
	`}`
	`};`

1.26.21. Frontend::Output::OutputFilterTextAutoLink###MSBulletins

Description	Value
Description:	Defines a filter for html output to add links behind MSBulletin numbers. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possiblity is to insert the link to the image.
Group:	Framework
SubGroup:	Core::Web

Description	Value
Valid:	0
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Output::OutputFilterTextAutoLink'}-
>{'MSBulletins'} =  {
  'RegExp' => [
    'MS[^A-Za-z]{0,5}(\\d\\d).?(\\d{2,4})'
  ],
  'Templates' => {
    'AgentTicketZoom' => '1'
  },
  'URL1' => {
    'Description' => 'Microsoft Technet',
    'Image' => 'http://www.microsoft.com/favicon.ico',
    'Target' => '_blank',
    'URL' => 'http://www.microsoft.com/technet/security/bulletin/
MS<MATCH1>-<MATCH2>.mspx'
  },
  'URL2' => {
    'Description' => 'Google',
    'Image' => 'http://www.google.de/favicon.ico',
    'Target' => '_blank',
    'URL' => 'http://google.com/search?q=MS<MATCH1>-<MATCH2>'
  }
};
```

1.26.22. Frontend::Output::OutputFilterTextAutoLink###Setting1

Description	Value
Description:	Define a filter for html output to add links behind a defined string. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possiblity is to insert the link to the image.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	

```
 $Self->{'Frontend::Output::OutputFilterTextAutoLink'}->{'Setting1'} =
  {
  'RegExp' => [
    'RegExp'
  ],
  'Templates' => {
    'AgentTicketZoom' => '1'
  },
  'URL1' => {
    'Description' => 'Description',
    'Image' => 'right-small.png',
    'Target' => '_blank',
    'URL' => 'URL'
  },
  'URL2' => {
    'Description' => 'Description',
    'Image' => 'Image',
    'Target' => '_blank',
    'URL' => 'URL'
  }
```

Description	Value
	`};`

1.26.23. Frontend::Output::OutputFilterTextAutoLink###Setting2

Description	Value
Description:	Defines a filter for html output to add links behind a defined string. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possiblity is to insert the link to the image.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Output::OutputFilterTextAutoLink'}->{'Setting2'} =
 {
 'RegExp' => [
   'RegExp'
 ],
 'Templates' => {
   'AgentTicketZoom' => '1'
 },
 'URL1' => {
   'Description' => 'Description',
   'Image' => 'right-small.png',
   'Target' => '_blank',
   'URL' => 'URL'
 },
 'URL2' => {
   'Description' => 'Description',
   'Image' => 'Image',
   'Target' => '_blank',
   'URL' => 'URL'
 },
 'URL3' => {
   'Description' => 'Description',
   'Image' => 'Image',
   'Target' => '_blank',
   'URL' => 'URL'
 }
};
```

1.26.24. Loader::Enabled::CSS

Description	Value
Description:	If enabled, OTRS will deliver all CSS files in minified form. WARNING: If you turn this off, there will likely be problems in IE 7, because it cannot load more than 32 CSS files.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'Loader::Enabled::CSS'} =  '1';
```

1.26.25. Loader::Enabled::JS

Description	Value
Description:	If enabled, OTRS will deliver all JavaScript files in minified form.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Loader::Enabled::JS'} =  '1';`

1.26.26. Loader::Agent::CommonCSS###000-Framework

Description	Value
Description:	List of CSS files to always be loaded for the agent interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'Loader::Agent::CommonCSS'}->{'000-Framework'} =  [
 'Core.Reset.css',
 'Core.Default.css',
 'Core.Header.css',
 'Core.OverviewControl.css',
 'Core.OverviewSmall.css',
 'Core.OverviewMedium.css',
 'Core.OverviewLarge.css',
 'Core.Footer.css',
 'Core.PageLayout.css',
 'Core.Form.css',
 'Core.Table.css',
 'Core.Widget.css',
 'Core.WidgetMenu.css',
 'Core.TicketDetail.css',
 'Core.Tooltip.css',
 'Core.Dialog.css',
 'Core.Print.css'
];
```

1.26.27. Loader::Agent::CommonCSS::IE8###000-Framework

Description	Value
Description:	List of IE8-specific CSS files to always be loaded for the agent interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'Loader::Agent::CommonCSS::IE8'}->{'000-Framework'} =  [
 'Core.OverviewSmall.IE8.css'
```

Description	Value
	`];`

1.26.28. Loader::Agent::CommonJS###000-Framework

Description	Value
Description:	List of JS files to always be loaded for the agent interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'Loader::Agent::CommonJS'}->{'000-Framework'} =  [
 'thirdparty/json/json2.js',
 'thirdparty/jquery-1.6.4/jquery.js',
 'thirdparty/jquery-ui-1.8.16/jquery-ui.js',
 'thirdparty/jquery-validate-1.9/jquery.validate.js',
 'thirdparty/stacktrace-0.3/stacktrace.js',
 'Core.JavaScriptEnhancements.js',
 'Core.Debug.js',
 'Core.Data.js',
 'Core.Config.js',
 'Core.Exception.js',
 'Core.JSON.js',
 'Core.AJAX.js',
 'Core.UI.js',
 'Core.UI.IE7Fixes.js',
 'Core.UI.Accordion.js',
 'Core.UI.Datepicker.js',
 'Core.UI.Resizable.js',
 'Core.UI.Table.js',
 'Core.UI.Accessibility.js',
 'Core.UI.RichTextEditor.js',
 'Core.UI.Dialog.js',
 'Core.UI.ActionRow.js',
 'Core.UI.Popup.js',
 'Core.Form.js',
 'Core.Form.ErrorTooltips.js',
 'Core.Form.Validate.js',
 'Core.App.js',
 'Core.Agent.js',
 'Core.Agent.Search.js'
];
```

1.26.29. Loader::Customer::CommonCSS###000-Framework

Description	Value
Description:	List of CSS files to always be loaded for the customer interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'Loader::Customer::CommonCSS'}->{'000-Framework'} =  [
 'Core.Reset.css',
 'Core.Default.css',
```

Description	Value

	`'Core.Form.css',`
	`'Core.Dialog.css',`
	`'Core.Tooltip.css',`
	`'Core.Login.css',`
	`'Core.Control.css',`
	`'Core.Table.css',`
	`'Core.TicketZoom.css',`
	`'Core.Print.css'`
	`];`

1.26.30. Loader::Customer::CommonCSS::IE6###000-Framework

Description	Value
Description:	List of IE6-specific CSS files to always be loaded for the customer interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Loader::Customer::CommonCSS::IE6'}->{'000-Framework'} = [`
	`'Core.IE6.css',`
	`'Core.Tooltip.IE6.css'`
	`];`

1.26.31. Loader::Customer::CommonCSS::IE7###000-Framework

Description	Value
Description:	List of IE7-specific CSS files to always be loaded for the customer interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Loader::Customer::CommonCSS::IE7'}->{'000-Framework'} = [`
	`'Core.IE7.css',`
	`'Core.Tooltip.IE7.css',`
	`'Core.Dialog.IE7.css'`
	`];`

1.26.32. Loader::Customer::CommonCSS::IE8###000-Framework

Description	Value
Description:	List of IE8-specific CSS files to always be loaded for the customer interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1

Description	Value
Config-Setting:	`$Self->{'Loader::Customer::CommonCSS::IE8'}->{'000-Framework'} = [];`

1.26.33. Loader::Customer::CommonJS###000-Framework

Description	Value
Description:	List of JS files to always be loaded for the customer interface.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Loader::Customer::CommonJS'}->{'000-Framework'} = ['thirdparty/jquery-1.6.4/jquery.js', 'thirdparty/jquery-validate-1.9/jquery.validate.js', 'thirdparty/jquery-ui-1.8.16/jquery-ui.js', 'thirdparty/stacktrace-0.3/stacktrace.js', 'Core.Debug.js', 'Core.Data.js', 'Core.Exception.js', 'Core.JavaScriptEnhancements.js', 'Core.Config.js', 'Core.App.js', 'Core.UI.js', 'Core.UI.IE7Fixes.js', 'Core.UI.Accessibility.js', 'Core.UI.Dialog.js', 'Core.UI.RichTextEditor.js', 'Core.UI.Datepicker.js', 'Core.UI.Popup.js', 'Core.Form.js', 'Core.Form.ErrorTooltips.js', 'Core.Form.Validate.js', 'Core.Customer.js'];`

1.26.34. Loader::Agent::DefaultSelectedSkin

Description	Value
Description:	The agent skin's InternalName which should be used in the agent interface. Please check the available skins in Frontend::Agent::Skins.
Group:	Framework
SubGroup:	Core::Web
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Loader::Agent::DefaultSelectedSkin'} = 'default';`

1.26.35. Loader::Customer::SelectedSkin::HostBased

Description	Value
Description:	It is possible to configure different skins, for example to distinguish between diferent customers, to be used on a per-domain basis within

Description	Value
	the application. Using a regular expression (regex), you can configure a Key/Content pair to match a domain. The value in "Key" should match the domain, and the value in "Content" should be a valid skin on your system. Please see the example entries for the proper form of the regex.
Group:	Framework
SubGroup:	Core::Web
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Loader::Customer::SelectedSkin::HostBased'} = { 'host1\\.example\\.com' => 'Someskin1', 'host2\\.example\\.com' => 'Someskin2' };`

1.27. Core::WebUserAgent

1.27.1. WebUserAgent::Timeout

Description	Value
Description:	Sets the timeout (in seconds) for http/ftp downloads.
Group:	Framework
SubGroup:	Core::WebUserAgent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'WebUserAgent::Timeout'} = '15';`

1.27.2. WebUserAgent::Proxy

Description	Value
Description:	Defines the connections for http/ftp, via a proxy.
Group:	Framework
SubGroup:	Core::WebUserAgent
Valid:	0
Required:	0
Config-Setting:	`$Self->{'WebUserAgent::Proxy'} = 'http://proxy.sn.no:8001/';`

1.28. Crypt::PGP

1.28.1. PGP

Description	Value
Description:	Enables PGP support. When PGP support is enabled for signing and securing mail, it is HIGHLY recommended that the web server be

Description	Value
	run as the OTRS user. Otherwise, there will be problems with the privileges when accessing .gnupg folder.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PGP'} = '0';`

1.28.2. PGP::Bin

Description	Value
Description:	Defines the path to PGP binary.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PGP::Bin'} = '/usr/bin/gpg';`

1.28.3. PGP::Options

Description	Value
Description:	Sets the options for PGP binary.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PGP::Options'} = '--homedir /opt/otrs/.gnupg/ --batch --no-tty --yes';`

1.28.4. PGP::Key::Password

Description	Value
Description:	Sets the password for private PGP key.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PGP::Key::Password'} = { '488A0B8F' => 'SomePassword', 'D2DF79FA' => 'SomePassword'`

Description	Value
	`};`

1.28.5. PGP::TrustedNetwork

Description	Value
Description:	Set this to yes if you trust in all your public and private pgp keys, even if they are not certified with a trusted signature.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	0
Config-Setting:	`$Self->{'PGP::TrustedNetwork'} =  '0';`

1.28.6. PGP::Log

Description	Value
Description:	Configure your own log text for PGP.
Group:	Framework
SubGroup:	Crypt::PGP
Valid:	1
Required:	1

Config-Setting:	`$Self->{'PGP::Log'} =  {`
	`  'BADSIG' => 'The PGP signature with the keyid has not been verified successfully.',`
	`  'ERRSIG' => 'It was not possible to check the PGP signature, this may be caused by a missing public key or an unsupported algorithm.',`
	`  'EXPKEYSIG' => 'The PGP signature was made by an expired key.',`
	`  'GOODSIG' => 'Good PGP signature.',`
	`  'KEYREVOKED' => 'The PGP signature was made by a revoked key, this could mean that the signature is forged.',`
	`  'NODATA' => 'No valid OpenPGP data found.',`
	`  'NO_PUBKEY' => 'No public key found.',`
	`  'REVKEYSIG' => 'The PGP signature was made by a revoked key, this could mean that the signature is forged.',`
	`  'SIGEXPIRED' => 'The PGP signature is expired.',`
	`  'SIG_ID' => 'Signature data.',`
	`  'TRUST_UNDEFINED' => 'This key is not certified with a trusted signature!.',`
	`  'VALIDSIG' => 'The PGP signature with the keyid is good.'`
	`};`

1.29. Crypt::SMIME

1.29.1. SMIME

Description	Value
Description:	Enables S/MIME support.
Group:	Framework
SubGroup:	Crypt::SMIME

Description	Value
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SMIME'} = '0';`

1.29.2. SMIME::Bin

Description	Value
Description:	Defines the path to open ssl binary. It may need a HOME env ($ENV{HOME} = '/var/lib/wwwrun';).
Group:	Framework
SubGroup:	Crypt::SMIME
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SMIME::Bin'} = '/usr/bin/openssl';`

1.29.3. SMIME::CertPath

Description	Value
Description:	Specifies the directory where SSL certificates are stored.
Group:	Framework
SubGroup:	Crypt::SMIME
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SMIME::CertPath'} = '/etc/ssl/certs';`

1.29.4. SMIME::PrivatePath

Description	Value
Description:	Specifies the directory where private SSL certificates are stored.
Group:	Framework
SubGroup:	Crypt::SMIME
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SMIME::PrivatePath'} = '/etc/ssl/private';`

1.30. Frontend::Admin::AdminCustomerUser

1.30.1. AdminCustomerUser::RunInitialWildcardSearch

Description	Value
Description:	Runs an initial wildcard search of the existing customer users when accessing the AdminCustomerUser module.

Description	Value
Group:	Framework
SubGroup:	Frontend::Admin::AdminCustomerUser
Valid:	1
Required:	1
Config-Setting:	`$Self->{'AdminCustomerUser::RunInitialWildcardSearch'} = '1';`

1.31. Frontend::Admin::ModuleRegistration

1.31.1. Frontend::Module###Admin

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'Admin'} =  {
 'Description' => 'Admin-Area',
 'Group' => [
   'admin'
 ],
 'Loader' => {
   'CSS' => [
     'Core.Agent.Admin.css'
   ],
   'JavaScript' => [
     'Core.Agent.Admin.SysConfig.js'
   ]
 },
 'NavBar' => [
   {
     'AccessKey' => 'a',
     'Block' => 'ItemArea',
     'Description' => '',
     'Link' => 'Action=Admin',
     'LinkOption' => '',
     'Name' => 'Admin',
     'NavBar' => 'Admin',
     'Prio' => '10000',
     'Type' => 'Menu'
   }
 ],
 'NavBarModule' => {
   'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin'
 },
 'NavBarName' => 'Admin',
 'Title' => ''
};
```

1.31.2. Frontend::Module###AdminInit

Description	Value
Description:	Frontend module registration for the agent interface.

Description	Value
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminInit'} = { 'Description' => 'Admin', 'Group' => [ 'admin' ], 'NavBarName' => '', 'Title' => 'Init' };`

1.31.3. Frontend::Module###AdminUser

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminUser'} = { 'Description' => 'Create and manage agents.', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Agent', 'Description' => 'Create and manage agents.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Agents', 'Prio' => '100' }, 'NavBarName' => 'Admin', 'Title' => 'Agents' };`

1.31.4. Frontend::Module###AdminGroup

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminGroup'} = { 'Description' => 'Admin', 'Group' => [`

Description	Value

	`'admin'`
	`],`
	`'NavBarModule' => {`
	`'Block' => 'Agent',`
	`'Description' => 'Create and manage groups.',`
	`'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`'Name' => 'Groups',`
	`'Prio' => '150'`
	`},`
	`'NavBarName' => 'Admin',`
	`'Title' => 'Groups'`
	`};`

1.31.5. Frontend::Module###AdminUserGroup

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminUserGroup'} = {`
	`'Description' => 'Admin',`
	`'Group' => [`
	`'admin'`
	`],`
	`'NavBarModule' => {`
	`'Block' => 'Agent',`
	`'Description' => 'Link agents to groups.',`
	`'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`'Name' => 'Agents <-> Groups',`
	`'Prio' => '200'`
	`},`
	`'NavBarName' => 'Admin',`
	`'Title' => 'Agents <-> Groups'`
	`};`

1.31.6. Frontend::Module###AdminCustomerUser

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminCustomerUser'} = {`
	`'Description' => 'Edit Customers',`
	`'Group' => [`
	`'admin',`
	`'users'`
	`],`
	`'GroupRo' => [`

Description	Value

	``
	    ''
	  ],
	  'Loader' => {
	    'JavaScript' => [
	      'Core.Agent.TicketAction.js'
	    ]
	  },
	  'NavBar' => [
	    {
	      'AccessKey' => 'c',
	      'Block' => 'ItemArea',
	      'Description' => '',
	      'Link' => 'Action=AdminCustomerUser;Nav=Agent',
	      'LinkOption' => '',
	      'Name' => 'Customers',
	      'NavBar' => 'Customers',
	      'Prio' => '9000',
	      'Type' => 'Menu'
	    }
	  ],
	  'NavBarModule' => {
	    'Block' => 'Customer',
	    'Description' => 'Create and manage customers.',
	    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	    'Name' => 'Customers',
	    'Prio' => '300'
	  },
	  'NavBarName' => 'Customers',
	  'Title' => 'Customers'
	};
	``

1.31.7. Frontend::Module###AdminCustomerCompany

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	```
	$Self->{'Frontend::Module'}->{'AdminCustomerCompany'} = {
	  'Description' => 'Admin',
	  'Group' => [
	    'admin',
	    'users'
	  ],
	  'GroupRo' => [
	    ''
	  ],
	  'NavBar' => [
	    {
	      'AccessKey' => 'c',
	      'Block' => 'ItemArea',
	      'Description' => '',
	      'Link' => 'Action=AdminCustomerCompany;Nav=Agent',
	      'LinkOption' => '',
	      'Name' => 'Companies',
	      'NavBar' => 'Companies',
	      'Prio' => '9100',
	      'Type' => 'Menu'
	``

Description	Value

	`    }`
	`  ],`
	`  'NavBarModule' => {`
	`    'Block' => 'Customer',`
	`    'Description' => 'Create and manage companies.',`
	`    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`    'Name' => 'Customer Companies',`
	`    'Prio' => '310'`
	`  },`
	`  'NavBarName' => 'Admin',`
	`  'Title' => 'Customer Companies'`
	`};`

1.31.8. Frontend::Module###AdminCustomerUserGroup

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminCustomerUserGroup'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'NavBarModule' => {`
	`    'Block' => 'Customer',`
	`    'Description' => 'Link customers to groups.',`
	`    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`    'Name' => 'Customers <-> Groups',`
	`    'Prio' => '400'`
	`  },`
	`  'NavBarName' => 'Admin',`
	`  'Title' => 'Customers <-> Groups'`
	`};`

1.31.9. Frontend::Module###AdminCustomerUserService

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminCustomerUserService'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'NavBarModule' => {`
	`    'Block' => 'Customer',`

Description	Value

	`'Description' => 'Link customers to services.',`
	`'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`'Name' => 'Customers <-> Services',`
	`'Prio' => '500'`
	`},`
	`'NavBarName' => 'Admin',`
	`'Title' => 'Customers <-> Services'`
	`};`

1.31.10. Frontend::Module###AdminRole

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminRole'} = {`
	`'Description' => 'Admin',`
	`'Group' => [`
	`'admin'`
	`],`
	`'NavBarModule' => {`
	`'Block' => 'Agent',`
	`'Description' => 'Create and manage roles.',`
	`'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`'Name' => 'Roles',`
	`'Prio' => '600'`
	`},`
	`'NavBarName' => 'Admin',`
	`'Title' => 'Roles'`
	`};`

1.31.11. Frontend::Module###AdminRoleUser

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminRoleUser'} = {`
	`'Description' => 'Admin',`
	`'Group' => [`
	`'admin'`
	`],`
	`'NavBarModule' => {`
	`'Block' => 'Agent',`
	`'Description' => 'Link agents to roles.',`
	`'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`'Name' => 'Agents <-> Roles',`
	`'Prio' => '700'`
	`},`

Description	Value

	`  'NavBarName' => 'Admin',`
	`  'Title' => 'Agents <-> Roles'`
	`};`

1.31.12. Frontend::Module###AdminRoleGroup

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminRoleGroup'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'NavBarModule' => {`
	`    'Block' => 'Agent',`
	`    'Description' => 'Link roles to groups.',`
	`    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`    'Name' => 'Roles <-> Groups',`
	`    'Prio' => '800'`
	`  },`
	`  'NavBarName' => 'Admin',`
	`  'Title' => 'Roles <-> Groups'`
	`};`

1.31.13. Frontend::Module###AdminSMIME

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminSMIME'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'NavBarModule' => {`
	`    'Block' => 'Email',`
	`    'Description' => 'Manage S/MIME certificates for email`
	` encryption.',`
	`    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`    'Name' => 'S/MIME Certificates',`
	`    'Prio' => '1100'`
	`  },`
	`  'NavBarName' => 'Admin',`
	`  'Title' => 'S/MIME Management'`
	`};`

1.31.14. Frontend::Module###AdminPGP

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AdminPGP'} =  {
 'Description' => 'Admin',
 'Group' => [
   'admin'
 ],
 'NavBarModule' => {
   'Block' => 'Email',
   'Description' => 'Manage PGP keys for email encryption.',
   'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
   'Name' => 'PGP Keys',
   'Prio' => '1200'
 },
 'NavBarName' => 'Admin',
 'Title' => 'PGP Key Management'
};
```

1.31.15. Frontend::Module###AdminMailAccount

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AdminMailAccount'} =  {
 'Description' => 'Admin',
 'Group' => [
   'admin'
 ],
 'NavBarModule' => {
   'Block' => 'Email',
   'Description' => 'Manage POP3 or IMAP accounts to fetch email
from.',
   'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
   'Name' => 'PostMaster Mail Accounts',
   'Prio' => '100'
 },
 'NavBarName' => 'Admin',
 'Title' => 'Mail Accounts'
};
```

1.31.16. Frontend::Module###AdminPostMasterFilter

Description	Value
Description:	Frontend module registration for the agent interface.

Description	Value
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AdminPostMasterFilter'} =  {
 'Description' => 'Admin',
 'Group' => [
   'admin'
 ],
 'NavBarModule' => {
   'Block' => 'Email',
   'Description' => 'Filter incoming emails.',
   'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
   'Name' => 'PostMaster Filters',
   'Prio' => '200'
 },
 'NavBarName' => 'Admin',
 'Title' => 'PostMaster Filters'
};
```

1.31.17. Frontend::Module###AdminEmail

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AdminEmail'} =  {
 'Description' => 'Admin',
 'Group' => [
   'admin'
 ],
 'NavBarModule' => {
   'Block' => 'System',
   'Description' => 'Send notifications to users.',
   'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
   'Name' => 'Admin Notification',
   'Prio' => '400'
 },
 'NavBarName' => 'Admin',
 'Title' => 'Admin Notification'
};
```

1.31.18. Frontend::Module###AdminSession

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1

Description	Value
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminSession'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'System', 'Description' => 'Manage existing sessions.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Session Management', 'Prio' => '500' }, 'NavBarName' => 'Admin', 'Title' => 'Session Management' };`

1.31.19. Frontend::Module###AdminPerformanceLog

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminPerformanceLog'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'Loader' => { 'CSS' => ['Core.Agent.Admin.PerformanceLog.css'] }, 'NavBarModule' => { 'Block' => 'System', 'Description' => 'View performance benchmark results.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Performance Log', 'Prio' => '550' }, 'NavBarName' => 'Admin', 'Title' => 'Performance Log' };`

1.31.20. Frontend::Module###AdminLog

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1

Description	Value
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminLog'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'System', 'Description' => 'View system log messages.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'System Log', 'Prio' => '600' }, 'NavBarName' => 'Admin', 'Title' => 'System Log' };`

1.31.21. Frontend::Module###AdminSelectBox

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminSelectBox'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'System', 'Description' => 'Execute SQL statements.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'SQL Box', 'Prio' => '700' }, 'NavBarName' => 'Admin', 'Title' => 'SQL Box' };`

1.31.22. Frontend::Module###AdminPackageManager

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminPackageManager'} = { 'Description' => 'Software Package Manager',`

Description	Value
	`'Group' => [ 'admin' ], 'NavBarModule' => { 'Block' => 'System', 'Description' => 'Update and extend your system with software packages.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Package Manager', 'Prio' => '1000' }, 'NavBarName' => 'Admin', 'Title' => 'Package Manager' };`

1.31.23. Frontend::Module###AdminGenericInterfaceDebugger

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminGenericInterfaceDebugger'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'Loader' => { 'CSS' => ['Core.Agent.Admin.GenericInterface.css'], 'CSS_IE7' => ['Core.Agent.Admin.GenericInterface.IE7.css'], 'JavaScript' => ['Core.Agent.Admin.GenericInterfaceDebugger.js'] }, 'Title' => 'GenericInterface Debugger GUI' };`

1.31.24. Frontend::Module###AdminGenericInterfaceWebservice

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminGenericInterfaceWebservice'} = { 'Description' => 'Admin',`

Description	Value

	``
	  'Group' => [
	    'admin'
	  ],
	  'Loader' => {
	    'CSS' => [
	      'Core.Agent.Admin.GenericInterface.css'
	    ],
	    'JavaScript' => [
	      'Core.Agent.Admin.GenericInterfaceWebservice.js'
	    ]
	  },
	  'NavBarModule' => {
	    'Block' => 'System',
	    'Description' => 'Create and manage web services.',
	    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
	    'Name' => 'Web Services',
	    'Prio' => '1000'
	  },
	  'NavBarName' => 'Admin',
	  'Title' => 'GenericInterface Web Service GUI'
	};
	``

1.31.25. Frontend::Module###AdminGenericInterfaceTransportHTTPSOAP

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	```
	 $Self->{'Frontend::Module'}-
	>{'AdminGenericInterfaceTransportHTTPSOAP'} = {
	  'Description' => 'Admin',
	  'Group' => [
	    'admin'
	  ],
	  'Loader' => {
	    'CSS' => [
	      'Core.Agent.Admin.GenericInterface.css'
	    ]
	  },
	  'Title' => 'GenericInterface TransportHTTPSOAP GUI'
	};
	``

1.31.26. Frontend::Module###AdminGenericInterfaceWebserviceHistory

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

Description	Value

	``
	$Self->{'Frontend::Module'}->{'AdminGenericInterfaceWebserviceHistory'} = {
	'Description' => 'Admin',
	'Group' => [
	'admin'
	],
	'Loader' => {
	'CSS' => [
	'Core.Agent.Admin.GenericInterface.css'
	],
	'JavaScript' => [
	'Core.Agent.Admin.GenericInterfaceWebserviceHistory.js'
	]
	},
	'Title' => 'GenericInterface Webservice History GUI'
	};
	``

1.31.27. Frontend::Module###AdminGenericInterfaceOperationDefault

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	```
	$Self->{'Frontend::Module'}->{'AdminGenericInterfaceOperationDefault'} = {
	'Description' => 'Admin',
	'Group' => [
	'admin'
	],
	'Loader' => {
	'CSS' => [
	'Core.Agent.Admin.GenericInterface.css'
	],
	'JavaScript' => [
	'Core.Agent.Admin.GenericInterfaceOperation.js'
	]
	},
	'Title' => 'GenericInterface Operation GUI'
	};
	``

1.31.28. Frontend::Module###AdminGenericInterfaceInvokerDefault

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	```
	$Self->{'Frontend::Module'}->{'AdminGenericInterfaceInvokerDefault'} = {
	``

Description	Value

	`'Description' => 'Admin',`
	`'Group' => [`
	`'admin'`
	`],`
	`'Loader' => {`
	`'CSS' => [`
	`'Core.Agent.Admin.GenericInterface.css'`
	`],`
	`'JavaScript' => [`
	`'Core.Agent.Admin.GenericInterfaceInvoker.js'`
	`]`
	`},`
	`'Title' => 'GenericInterface Invoker GUI'`
	`};`

1.31.29. Frontend::Module###AdminGenericInterfaceMappingSimple

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminGenericInterfaceMappingSimple'} =`
	`{`
	`'Description' => 'Admin',`
	`'Group' => [`
	`'admin'`
	`],`
	`'Loader' => {`
	`'CSS' => [`
	`'Core.Agent.Admin.GenericInterface.css'`
	`],`
	`'CSS_IE7' => [`
	`'Core.Agent.Admin.GenericInterface.IE7.css'`
	`],`
	`'JavaScript' => [`
	`'Core.Agent.Admin.GenericInterfaceMappingSimple.js'`
	`]`
	`},`
	`'Title' => 'GenericInterface Webservice Mapping GUI'`
	`};`

1.32. Frontend::Agent

1.32.1. AgentLogo

Description	Value
Description:	The logo shown in the header of the agent interface. The URL to the image can be a relative URL to the skin image directory, or a full URL to a remote web server.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1

Description	Value
Required:	0
Config-Setting:	`$Self->{'AgentLogo'} = { 'StyleHeight' => '67px', 'StyleRight' => '38px', 'StyleTop' => '-4px', 'StyleWidth' => '244px', 'URL' => 'skins/Agent/default/img/logo_bg.png' };`

1.32.2. AgentLoginLogo

Description	Value
Description:	The logo shown on top of the login box of the agent interface. The URL to the image must be relative URL to the skin image directory.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	`$Self->{'AgentLoginLogo'} = { 'StyleHeight' => '100px', 'URL' => 'skins/Agent/default/img/loginlogo_default.png' };`

1.32.3. LoginURL

Description	Value
Description:	Defines an alternate URL, where the login link refers to.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	`$Self->{'LoginURL'} = 'http://host.example.com/login.html';`

1.32.4. LogoutURL

Description	Value
Description:	Defines an alternate URL, where the logout link refers to.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	`$Self->{'LogoutURL'} = 'http://host.example.com/thanks-for-using-otrs.html';`

1.32.5. PreApplicationModule###AgentInfo

Description	Value
Description:	Defines a useful module to load specific user options or to display news.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	`$Self->{'PreApplicationModule'}->{'AgentInfo'} = 'Kernel::Modules::AgentInfo';`

1.32.6. InfoKey

Description	Value
Description:	Defines the key to be checked with Kernel::Modules::AgentInfo module. If this user preferences key is true, the message is accepted by the system.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	`$Self->{'InfoKey'} =  'wpt22';`

1.32.7. InfoFile

Description	Value
Description:	File that is displayed in the Kernel::Modules::AgentInfo module, if located under Kernel/Output/HTML/Standard/AgentInfo.dtl.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	`$Self->{'InfoFile'} =  'AgentInfo';`

1.32.8. LostPassword

Description	Value
Description:	Activates lost password feature for agents, in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1

Description	Value
Required:	1
Config-Setting:	`$Self->{'LostPassword'} = '1';`

1.32.9. ShowMotd

Description	Value
Description:	Shows the message of the day on login screen of the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'ShowMotd'} = '0';`

1.32.10. NotificationSubjectLostPasswordToken

Description	Value
Description:	Defines the subject for notification mails sent to agents, with token about new requested password.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'NotificationSubjectLostPasswordToken'} = 'New OTRS password request';`

1.32.11. NotificationBodyLostPasswordToken

Description	Value
Description:	Defines the body text for notification mails sent to agents, with token about new requested password (after using this link the new password will be sent).
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1

Config-Setting:	`$Self->{'NotificationBodyLostPasswordToken'} = 'Hi <OTRS_USERFIRSTNAME>,`
	
	`You or someone impersonating you has requested to change your OTRS password.`
	
	`If you want to do this, click on the link below. You will receive another email containing the password.`

Description	Value
	<OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/ <OTRS_CONFIG_ScriptAlias>index.pl? Action=LostPassword;Token=<OTRS_TOKEN> If you did not request a new password, please ignore this email. ';

1.32.12. NotificationSubjectLostPassword

Description	Value
Description:	Defines the subject for notification mails sent to agents, about new password.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	$Self->{'NotificationSubjectLostPassword'} = 'New OTRS password';

1.32.13. NotificationBodyLostPassword

Description	Value
Description:	Defines the body text for notification mails sent to agents, about new password (after using this link the new password will be sent).
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	$Self->{'NotificationBodyLostPassword'} = 'Hi <OTRS_USERFIRSTNAME>, Here\'s your new OTRS password. New password: <OTRS_NEWPW> You can log in via the following URL: <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/ <OTRS_CONFIG_ScriptAlias>index.pl ';

1.32.14. OpenMainMenuOnHover

Description	Value
Description:	If enabled, the first level of the main menu opens on mouse hover (instead of click only).
Group:	Framework
SubGroup:	Frontend::Agent

Description	Value
Valid:	1
Required:	1
Config-Setting:	`$Self->{'OpenMainMenuOnHover'} =  '0';`

1.32.15. Loader::Agent::Skin###000-default

Description	Value
Description:	Default skin for interface.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Loader::Agent::Skin'}->{'000-default'} =  { 'Description' => 'This is the default orange - black skin.', 'HomePage' => 'www.otrs.org', 'InternalName' => 'default', 'VisibleName' => 'Default' };`

1.32.16. Loader::Agent::Skin###001-ivory

Description	Value
Description:	Balanced white skin by Felix Niklas.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Loader::Agent::Skin'}->{'001-ivory'} =  { 'Description' => 'Balanced white skin by Felix Niklas', 'HomePage' => 'www.felixniklas.de', 'InternalName' => 'ivory', 'VisibleName' => 'Ivory' };`

1.32.17. Loader::Agent::Skin###001-slim

Description	Value
Description:	Experimental "Slim" skin which tries to save screen space for power users.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Loader::Agent::Skin'}->{'001-slim'} =  {`

Description	Value
	`'Description' => 'Experimental "Slim" skin which tries to save screen space for power users.', 'HomePage' => 'www.otrs.org', 'InternalName' => 'slim', 'VisibleName' => 'Slim' };`

1.32.18. Loader::Agent::DefaultSelectedSkin::HostBased

Description	Value
Description:	It is possible to configure different skins, for example to distinguish between diferent agents, to be used on a per-domain basis within the application. Using a regular expression (regex), you can configure a Key/Content pair to match a domain. The value in "Key" should match the domain, and the value in "Content" should be a valid skin on your system. Please see the example entries for the proper form of the regex.
Group:	Framework
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Loader::Agent::DefaultSelectedSkin::HostBased'} = { 'host1\\.example\\.com' => 'SomeSkin1', 'host2\\.example\\.com' => 'SomeSkin2' };`

1.33. Frontend::Agent::Dashboard

1.33.1. DashboardBackend###0000-ProductNotify

Description	Value
Description:	Defines the parameters for the dashboard backend. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" defines the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	`$Self->{'DashboardBackend'}->{'0000-ProductNotify'} = { 'Block' => 'ContentLarge', 'CacheTTLLocal' => '1440', 'Default' => '1', 'Description' => 'News about OTRS releases!', 'Group' => 'admin', 'Module' => 'Kernel::Output::HTML::DashboardProductNotify', 'Title' => 'Product News', 'URL' => 'http://otrs.org/product.xml' };`

1.33.2. DashboardBackend###0400-UserOnline

Description	Value
Description:	Defines the parameters for the dashboard backend. "Limit" defines the number of entries displayed by default. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" defines the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	`$Self->{'DashboardBackend'}->{'0400-UserOnline'} = { 'Block' => 'ContentSmall', 'CacheTTLLocal' => '5', 'Default' => '0', 'Description' => '', 'Filter' => 'Agent', 'Group' => '', 'IdleMinutes' => '60', 'Limit' => '10', 'Module' => 'Kernel::Output::HTML::DashboardUserOnline', 'ShowEmail' => '1', 'SortBy' => 'UserLastname', 'Title' => 'Online' };`

1.33.3. DashboardBackend###0410-RSS

Description	Value
Description:	Defines the parameters for the dashboard backend. "Limit" defines the number of entries displayed by default. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTL" indicates the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	`$Self->{'DashboardBackend'}->{'0410-RSS'} = { 'Block' => 'ContentSmall', 'CacheTTL' => '360', 'Default' => '1', 'Description' => '', 'Group' => '', 'Limit' => '6', 'Module' => 'Kernel::Output::HTML::DashboardRSS', 'Title' => 'OTRS News', 'URL' => 'http://www.otrs.com/en/rss.xml', 'URL_de' => 'http://www.otrs.com/de/rss.xml',`

Description	Value

	`'URL_es' => 'http://www.otrs.com/es/rss.xml',`
	`'URL_nl' => 'http://www.otrs.com/nl/rss.xml',`
	`'URL_ru' => 'http://www.otrs.com/ru/rss.xml',`
	`'URL_zh' => 'http://www.otrs.com/cn/rss.xml'`
	`};`

1.33.4. DashboardBackend###0200-Image

Description	Value
Description:	Defines the parameters for the dashboard backend. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTL" indicates the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	0
Required:	0

Config-Setting:	`$Self->{'DashboardBackend'}->{'0200-Image'} = {`
	`'Block' => 'ContentLarge',`
	`'Default' => '1',`
	`'Description' => 'Some picture description!',`
	`'Group' => '',`
	`'Height' => '140',`
	`'Link' => 'http://otrs.org/',`
	`'LinkTitle' => 'http://otrs.org/',`
	`'Module' => 'Kernel::Output::HTML::DashboardImage',`
	`'Title' => 'A picture',`
	`'URL' => 'http://www.otrs.com/uploads/pics/jointhecommunity_02.jpg',`
	`'Width' => '198'`
	`};`

1.33.5. DashboardBackend###0210-MOTD

Description	Value
Description:	Shows the message of the day (MOTD) in the agent dashboard. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	0
Required:	0

Config-Setting:	`$Self->{'DashboardBackend'}->{'0210-MOTD'} = {`
	`'Block' => 'ContentLarge',`
	`'Default' => '1',`
	`'Group' => '',`
	`'Module' => 'Kernel::Output::HTML::DashboardMOTD',`
	`'Title' => 'Message of the Day'`
	`};`

1.33.6. DashboardBackend###0300-IFrame

Description	Value
Description:	Defines the parameters for the dashboard backend. "Group" is used to restrict access to the plugin (e. g. Group: admin;group1;group2;). "Default" indicates if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTL" indicates the cache expiration period in minutes for the plugin.
Group:	Framework
SubGroup:	Frontend::Agent::Dashboard
Valid:	0
Required:	0
Config-Setting:	

```
$Self->{'DashboardBackend'}->{'0300-IFrame'} =  {
 'Align' => 'left',
 'Block' => 'ContentLarge',
 'Default' => '1',
 'Description' => 'Some description!',
 'Frameborder' => '1',
 'Group' => '',
 'Height' => '800',
 'Link' => 'http://otrs.org/',
 'LinkTitle' => 'OTRS.org/',
 'Marginheight' => '5',
 'Marginwidth' => '5',
 'Module' => 'Kernel::Output::HTML::DashboardIFrame',
 'Scrolling' => 'auto',
 'Title' => 'A Website',
 'URL' => 'http://www.otrs.org/',
 'Width' => '1024'
};
```

1.34. Frontend::Agent::LinkObject

1.34.1. Frontend::AgentLinkObject::WildcardSearch

Description	Value
Description:	Starts a wildcard search of the active object after the link object mask is started.
Group:	Framework
SubGroup:	Frontend::Agent::LinkObject
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'Frontend::AgentLinkObject::WildcardSearch'} =  '0';
```

1.35. Frontend::Agent::ModuleMetaHead

1.35.1. Frontend::HeaderMetaModule###100-Refresh

Description	Value
Description:	Defines the module to generate html refresh headers of html sites.
Group:	Framework

Description	Value
SubGroup:	Frontend::Agent::ModuleMetaHead
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::HeaderMetaModule'}->{'100-Refresh'} = { 'Module' => 'Kernel::Output::HTML::HeaderMetaRefresh' };`

1.36. Frontend::Agent::ModuleNotify

1.36.1. Frontend::NotifyModule###200-UID-Check

Description	Value
Description:	Defines the module to display a notification in the agent interface, if the system is used by the admin user (normally you shouldn't work as admin).
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'200-UID-Check'} = { 'Module' => 'Kernel::Output::HTML::NotificationUIDCheck' };`

1.36.2. Frontend::NotifyModule###300-ShowAgentOnline

Description	Value
Description:	Defines the module that shows all the currently logged in agents in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'300-ShowAgentOnline'} = { 'IdleMinutes' => '60', 'Module' => 'Kernel::Output::HTML::NotificationAgentOnline', 'ShowEmail' => '1' };`

1.36.3. Frontend::NotifyModule###400-ShowCustomerOnline

Description	Value
Description:	Defines the module that shows all the currently loged in customers in the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify

Description	Value
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'400-ShowCustomerOnline'} = { 'IdleMinutes' => '60', 'Module' => 'Kernel::Output::HTML::NotificationCustomerOnline', 'ShowEmail' => '1' };`

1.36.4. Frontend::NotifyModule###500-OutofOffice-Check

Description	Value
Description:	Defines the module to display a notification in the agent interface, if the agent is logged in while having out-of-office active.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'500-OutofOffice-Check'} = { 'Module' => 'Kernel::Output::HTML::NotificationOutofOfficeCheck' };`

1.36.5. Frontend::NotifyModule###900-Generic

Description	Value
Description:	Defines the module that shows a generic notification in the agent interface. Either "Text" - if configured - or the contents of "File" will be displayed.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'900-Generic'} = { 'File' => '<OTRS_CONFIG_Home>/var/notify.txt', 'Link' => 'http://www.otrs.com', 'Module' => 'Kernel::Output::HTML::NotificationGeneric', 'Priority' => 'Warning', 'Text' => 'The OTRS Website' };`

1.37. Frontend::Agent::ModuleRegistration

1.37.1. Frontend::Module###Logout

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration

Description	Value
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'Logout'} = { 'Description' => 'Logout', 'NavBarName' => '', 'Title' => '' };`

1.37.2. Frontend::Module###AgentDashboard

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentDashboard'} = { 'Description' => 'Agent Dashboard', 'Loader' => { 'JavaScript' => ['thirdparty/flot-0.7/excanvas.js', 'thirdparty/flot-0.7/jquery.flot.js', 'Core.UI.Chart.js', 'Core.UI.DnD.js', 'Core.Agent.Dashboard.js'] }, 'NavBar' => [{ 'AccessKey' => 'd', 'Block' => 'ItemArea', 'Description' => '', 'Link' => 'Action=AgentDashboard', 'LinkOption' => '', 'Name' => 'Dashboard', 'NavBar' => 'Dashboard', 'Prio' => '50', 'Type' => 'Menu' }], 'NavBarName' => 'Dashboard', 'Title' => '' };`

1.37.3. Frontend::Module###AgentPreferences

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentPreferences'} = { 'Description' => 'Agent Preferences', 'NavBarName' => 'Preferences', 'Title' => '' };`

1.37.4. Frontend::Module###PictureUpload

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::Module'}->{'PictureUpload'} = { 'Description' => 'Picture upload module', 'NavBarName' => 'Ticket', 'Title' => 'Picture-Upload' };`

1.37.5. Frontend::Module###AgentSpelling

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentSpelling'} = { 'Description' => 'Spell checker', 'Loader' => { 'JavaScript' => [ 'Core.Agent.TicketAction.js' ] }, 'NavBarName' => '', 'Title' => 'Spell Checker' };`

1.37.6. Frontend::Module###SpellingInline

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	`$Self->{'Frontend::Module'}->{'SpellingInline'} = { 'Description' => 'Spell checker', 'NavBarName' => '', 'Title' => 'Spell Checker' };`

1.37.7. Frontend::Module###AgentBook

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentBook'} = { 'Description' => 'Address book of CustomerUser sources', 'Loader' => { 'JavaScript' => [ 'Core.Agent.CustomerSearch.js', 'Core.Agent.TicketAction.js' ] }, 'NavBarName' => '', 'Title' => 'Address Book' };`

1.37.8. Frontend::Module###AgentLinkObject

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentLinkObject'} = { 'Description' => 'Link Object', 'NavBarName' => '', 'Title' => 'Link Object' };`

1.37.9. Frontend::Module###AgentInfo

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Description	Value
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentInfo'} = { 'Description' => 'Generic Info module', 'NavBarName' => '', 'Title' => 'Info' };`

1.37.10. Frontend::Module###AgentSearch

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentSearch'} = { 'Description' => 'Global Search Module', 'NavBarName' => '', 'Title' => 'Search' };`

1.37.11. CustomerFrontend::Module###SpellingInline

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'SpellingInline'} = { 'Description' => 'Spell checker', 'NavBarName' => '', 'Title' => 'Spell Checker' };`

1.37.12. Frontend::Module###AgentHTMLReference

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentHTMLReference'} = { 'Description' => 'HTML Reference', 'Group' => [`

Description	Value

	`    'users'`
	`  ],`
	`  'GroupRo' => [`
	`    'users'`
	`  ],`
	`  'Loader' => {`
	`    'CSS' => [`
	`      'Core.Agent.HTMLReference.css'`
	`    ]`
	`  },`
	`  'NavBarName' => '',`
	`  'Title' => 'HTML Reference'`
	`};`

1.37.13. Frontend::Module###AgentStats

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentStats'} =  {`
	`  'Description' => 'Stats',`
	`  'Group' => [`
	`    'stats'`
	`  ],`
	`  'GroupRo' => [`
	`    'stats'`
	`  ],`
	`  'Loader' => {`
	`    'JavaScript' => [`
	`      'Core.Agent.Stats.js'`
	`    ]`
	`  },`
	`  'NavBar' => [`
	`    {`
	`      'AccessKey' => '',`
	`      'Block' => 'ItemArea',`
	`      'Description' => '',`
	`      'Link' => 'Action=AgentStats;Subaction=Overview',`
	`      'LinkOption' => '',`
	`      'Name' => 'Statistics',`
	`      'NavBar' => 'Stats',`
	`      'Prio' => '8500',`
	`      'Type' => 'Menu'`
	`    },`
	`    {`
	`      'AccessKey' => '',`
	`      'Block' => '',`
	`      'Description' => 'Overview',`
	`      'GroupRo' => [`
	`        'stats'`
	`      ],`
	`      'Link' => 'Action=AgentStats;Subaction=Overview',`
	`      'LinkOption' => '',`
	`      'Name' => 'Overview',`
	`      'NavBar' => 'Stats',`
	`      'Prio' => '100',`

Description	Value

	``
	      'Type' => ''
	    },
	    {
	      'AccessKey' => '',
	      'Block' => '',
	      'Description' => 'New',
	      'Group' => [
	        'stats'
	      ],
	      'Link' => 'Action=AgentStats;Subaction=Add',
	      'LinkOption' => '',
	      'Name' => 'New',
	      'NavBar' => 'Stats',
	      'Prio' => '200',
	      'Type' => ''
	    },
	    {
	      'AccessKey' => '',
	      'Block' => '',
	      'Description' => 'Import',
	      'Group' => [
	        'stats'
	      ],
	      'Link' => 'Action=AgentStats;Subaction=Import',
	      'LinkOption' => '',
	      'Name' => 'Import',
	      'NavBar' => 'Stats',
	      'Prio' => '300',
	      'Type' => ''
	    }
	  ],
	  'NavBarName' => 'Stats',
	  'Title' => 'Stats'
	};
	``

1.38. Frontend::Agent::NavBarModule

1.38.1. Frontend::NavBarModule###6-CustomerCompany

Description	Value
Description:	Frontend module registration (disable company link if no company feature is used).
Group:	Framework
SubGroup:	Frontend::Agent::NavBarModule
Valid:	1
Required:	0

Config-Setting:	```
	$Self->{'Frontend::NavBarModule'}->{'6-CustomerCompany'} = {
	  'Module' => 'Kernel::Output::HTML::NavBarCustomerCompany'
	};
	``

1.39. Frontend::Agent::Preferences

1.39.1. PreferencesTableValue

Description	Value
Description:	Defines the name of the column to store the data in the preferences table.

Description	Value
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PreferencesTableValue'} = 'preferences_value';`

1.39.2. PreferencesTableUserID

Description	Value
Description:	Defines the name of the column to store the user identifier in the preferences table.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PreferencesTableUserID'} = 'user_id';`

1.39.3. PreferencesView

Description	Value
Description:	Sets the display order of the different items in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PreferencesView'} = [ 'User Profile', 'Email Settings', 'Other Settings' ];`

1.39.4. PreferencesGroups###Password

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	`$Self->{'PreferencesGroups'}->{'Password'} = { 'Active' => '1', 'Area' => 'Agent',`

Description	Value

	`  'Column' => 'User Profile',`
	`  'Label' => 'Change password',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesPassword',`
	`  'PasswordMaxLoginFailed' => '0',`
	`  'PasswordMin2Characters' => '0',`
	`  'PasswordMin2Lower2UpperCharacters' => '0',`
	`  'PasswordMinSize' => '0',`
	`  'PasswordNeedDigit' => '0',`
	`  'PasswordRegExp' => '',`
	`  'Prio' => '0500'`
	`};`

1.39.5. PreferencesGroups###SpellDict

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view. Take care to maintain the dictionaries installed in the system in the data section.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	0
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'SpellDict'} =  {`
	`  'Active' => '1',`
	`  'Column' => 'User Profile',`
	`  'Data' => {`
	`    'deutsch' => 'Deutsch',`
	`    'english' => 'English'`
	`  },`
	`  'DataSelected' => 'english',`
	`  'Key' => 'Default spelling dictionary',`
	`  'Label' => 'Spelling Dictionary',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',`
	`  'PrefKey' => 'UserSpellDict',`
	`  'Prio' => '2000'`
	`};`

1.39.6. PreferencesGroups###Comment

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'Comment'} =  {`
	`  'Active' => '0',`
	`  'Block' => 'Input',`
	`  'Column' => 'Other Settings',`
	`  'Data' => '$Env{"UserComment"}',`
	`  'Key' => 'Comment',`

Description	Value

	`'Label' => 'Comment',`
	`'Module' => 'Kernel::Output::HTML::PreferencesGeneric',`
	`'PrefKey' => 'UserComment',`
	`'Prio' => '6000'`
	`};`

1.39.7. PreferencesGroups###Language

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'Language'} = {`
	`'Active' => '1',`
	`'Column' => 'User Profile',`
	`'Key' => 'Frontend language',`
	`'Label' => 'Language',`
	`'Module' => 'Kernel::Output::HTML::PreferencesLanguage',`
	`'PrefKey' => 'UserLanguage',`
	`'Prio' => '1000'`
	`};`

1.39.8. PreferencesGroups###Skin

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'Skin'} = {`
	`'Active' => '1',`
	`'Column' => 'User Profile',`
	`'Key' => 'Wear this frontend skin',`
	`'Label' => 'Skin',`
	`'Module' => 'Kernel::Output::HTML::PreferencesSkin',`
	`'PrefKey' => 'UserSkin',`
	`'Prio' => '2000'`
	`};`

1.39.9. PreferencesGroups###Theme

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework

Description	Value
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	`$Self->{'PreferencesGroups'}->{'Theme'} = { 'Active' => '1', 'Column' => 'User Profile', 'Key' => 'Frontend theme', 'Label' => 'Theme', 'Module' => 'Kernel::Output::HTML::PreferencesTheme', 'PrefKey' => 'UserTheme', 'Prio' => '3000' };`

1.39.10. PreferencesGroups###OutOfOffice

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	`$Self->{'PreferencesGroups'}->{'OutOfOffice'} = { 'Active' => '1', 'Block' => 'OutOfOffice', 'Column' => 'User Profile', 'Key' => '', 'Label' => 'Out Of Office Time', 'Module' => 'Kernel::Output::HTML::PreferencesOutOfOffice', 'PrefKey' => 'UserOutOfOffice', 'Prio' => '4000' };`

1.39.11. PreferencesGroups###TimeZone

Description	Value
Description:	Defines the config parameters of this item, to be shown in the preferences view.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	`$Self->{'PreferencesGroups'}->{'TimeZone'} = { 'Active' => '1', 'Column' => 'User Profile', 'Key' => 'Time Zone', 'Label' => 'Time Zone', 'Module' => 'Kernel::Output::HTML::PreferencesTimeZone', 'PrefKey' => 'UserTimeZone', 'Prio' => '5000'`

Description	Value
	`};`

1.39.12. PreferencesGroups###CSVSeparator

Description	Value
Description:	Gives end users the possibility to override the separator character for CSV files, defined in the translation files.
Group:	Framework
SubGroup:	Frontend::Agent::Preferences
Valid:	0
Required:	0
Config-Setting:	

```
$Self->{'PreferencesGroups'}->{'CSVSeparator'} =  {
 'Active' => '1',
 'Column' => 'Other Settings',
 'Data' => {
   '' => '',
   ',' => ',',
   ';' => ';',
   '\\t' => 'tab',
   '|' => '|'
 },
 'DataSelected' => '0',
 'Desc' => 'Select the separator character used in CSV files (stats
and searches). If you don\'t select a separator here, the default
separator for your language will be used.',
 'Key' => 'CSV Separator',
 'Label' => 'CSV Separator',
 'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
 'PrefKey' => 'UserCSVSeparator',
 'Prio' => '4000'
};
```

1.40. Frontend::Agent::SearchRouter

1.40.1. Frontend::SearchDefault

Description	Value
Description:	Search backend default router.
Group:	Framework
SubGroup:	Frontend::Agent::SearchRouter
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::SearchDefault'} =
'Action=AgentTicketSearch;Subaction=AJAX';
```

1.41. Frontend::Agent::Stats

1.41.1. Stats::SearchPageShown

Description	Value
Description:	Defines the default maximum number of search results shown on the overview page.

Description	Value
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::SearchPageShown'} = '20';`

1.41.2. Stats::DefaultSelectedDynamicObject

Description	Value
Description:	Defines the default selection at the drop down menu for dynamic objects (Form: Common Specification).
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::DefaultSelectedDynamicObject'} = 'Ticket';`

1.41.3. Stats::DefaultSelectedPermissions

Description	Value
Description:	Defines the default selection at the drop down menu for permissions (Form: Common Specification).
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::DefaultSelectedPermissions'} = [ 'stats' ];`

1.41.4. Stats::DefaultSelectedFormat

Description	Value
Description:	Defines the default selection at the drop down menu for stats format (Form: Common Specification). Please insert the format key (see Stats::Format).
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	

Description	Value

	`$Self->{'Stats::DefaultSelectedFormat'} = [`
	`'Print',`
	`'CSV'`
	`];`

1.41.5. Stats::SearchLimit

Description	Value
Description:	Defines the search limit for the stats.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::SearchLimit'} = '500';`

1.41.6. Stats::Format

Description	Value
Description:	Defines all the possible stats output formats.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Stats::Format'} = {`
	`'CSV' => 'CSV',`
	`'GD::Graph::area' => 'graph-area',`
	`'GD::Graph::bars' => 'graph-bars',`
	`'GD::Graph::hbars' => 'graph-hbars',`
	`'GD::Graph::lines' => 'graph-lines',`
	`'GD::Graph::linespoints' => 'graph-lines-points',`
	`'GD::Graph::pie' => 'graph-pie',`
	`'GD::Graph::points' => 'graph-points',`
	`'Print' => 'Print'`
	`};`

1.41.7. Stats::GraphSize

Description	Value
Description:	Sets the size of the statistic graph.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::GraphSize'} = {` `'1200x800' => '1200x800',`

Description	Value

	`'1600x1200' => '1600x1200',`
	`'800x600' => '800x600'`
	`};`

1.41.8. Stats::TimeType

Description	Value
Description:	Sets the time type which should be shown.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Stats::TimeType'} = 'Extended';`

1.41.9. Stats::ExchangeAxis

Description	Value
Description:	Allows agents to exchange the axis of a stat if they generate one.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Stats::ExchangeAxis'} = '0';`

1.41.10. Stats::UseAgentElementInStats

Description	Value
Description:	Allows agents to generate individual-related stats.
Group:	Framework
SubGroup:	Frontend::Agent::Stats
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Stats::UseAgentElementInStats'} = '0';`

1.41.11. Stats::CustomerIDAsMultiSelect

Description	Value
Description:	Shows all the customer identifiers in a multi-select field (not useful if you have a lot of customer identifiers).
Group:	Framework
SubGroup:	Frontend::Agent::Stats

Description	Value
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Stats::CustomerIDAsMultiSelect'} = '1';`

1.42. Frontend::Customer

1.42.1. CustomerHeadline

Description	Value
Description:	The headline shown in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerHeadline'} = 'Example Company Support';`

1.42.2. CustomerLogo

Description	Value
Description:	The logo shown in the header of the customer interface. The URL to the image can be a relative URL to the skin image directory, or a full URL to a remote web server.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'CustomerLogo'} = { 'StyleHeight' => '50px', 'StyleRight' => '25px', 'StyleTop' => '2px', 'StyleWidth' => '135px', 'URL' => 'skins/Customer/default/img/logo.png' };`

1.42.3. CustomerPanelUserID

Description	Value
Description:	Defines the user identifier for the customer panel.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelUserID'} = '1';`

1.42.4. CustomerGroupSupport

Description	Value
Description:	Activates support for customer groups.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerGroupSupport'} = '0';`

1.42.5. CustomerGroupAlwaysGroups

Description	Value
Description:	Defines the groups every customer user will be in (if CustomerGroupSupport is enabled and you don't want to manage every user for these groups).
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerGroupAlwaysGroups'} = [ 'users' ];`

1.42.6. CustomerPanelLoginURL

Description	Value
Description:	Defines an alternate login URL for the customer panel..
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'CustomerPanelLoginURL'} = 'http://host.example.com/cgi-bin/login.pl';`

1.42.7. CustomerPanelLogoutURL

Description	Value
Description:	Defines an alternate logout URL for the customer panel.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0

Description	Value
Config-Setting:	`$Self->{'CustomerPanelLogoutURL'} = 'http://host.example.com/cgi-bin/login.pl';`

1.42.8. Frontend::CustomerUser::Item###1-GoogleMaps

Description	Value
Description:	Defines a customer item, which generates a google maps icon at the end of a customer info block.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::CustomerUser::Item'}->{'1-GoogleMaps'} = { 'Attributes' => 'UserStreet;UserCity;UserCountry;', 'CSS' => 'Core.Agent.CustomerUser.GoogleMaps.css', 'CSSClass' => 'GoogleMaps', 'Module' => 'Kernel::Output::HTML::CustomerUserGeneric', 'Required' => 'UserStreet;UserCity;', 'Target' => '_blank', 'Text' => 'Location', 'URL' => 'http://maps.google.com/maps?z=7&q=' };`

1.42.9. Frontend::CustomerUser::Item###2-Google

Description	Value
Description:	Defines a customer item, which generates a google icon at the end of a customer info block.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::CustomerUser::Item'}->{'2-Google'} = { 'Attributes' => 'UserFirstname;UserLastname;', 'CSS' => 'Core.Agent.CustomerUser.Google.css', 'CSSClass' => 'Google', 'Module' => 'Kernel::Output::HTML::CustomerUserGeneric', 'Required' => 'UserFirstname;UserLastname;', 'Target' => '_blank', 'Text' => 'Google', 'URL' => 'http://google.com/search?q=' };`

1.42.10. Frontend::CustomerUser::Item###2-LinkedIn

Description	Value
Description:	Defines a customer item, which generates a LinkedIn icon at the end of a customer info block.

Description	Value
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::CustomerUser::Item'}->{'2-LinkedIn'} = { 'Attributes' => 'UserFirstname;UserLastname;', 'CSS' => 'Core.Agent.CustomerUser.LinkedIn.css', 'CSSClass' => 'LinkedIn', 'Module' => 'Kernel::Output::HTML::CustomerUserGeneric', 'Required' => 'UserFirstname;UserLastname;', 'Target' => '_blank', 'Text' => 'LinkedIn', 'URL' => 'http://www.linkedin.com/commonSearch?type=people&keywords=' };`

1.42.11. Frontend::CustomerUser::Item###3-XING

Description	Value
Description:	Defines a customer item, which generates a XING icon at the end of a customer info block.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::CustomerUser::Item'}->{'3-XING'} = { 'Attributes' => 'UserFirstname;UserLastname;', 'CSS' => 'Core.Agent.CustomerUser.Xing.css', 'CSSClass' => 'Xing', 'Module' => 'Kernel::Output::HTML::CustomerUserGeneric', 'Required' => 'UserFirstname;UserLastname;', 'Target' => '_blank', 'Text' => 'XING', 'URL' => 'https://www.xing.com/app/search?op=search;keywords=' };`

1.42.12. CustomerPanelPreApplicationModule###CustomerAccept

Description	Value
Description:	This module and its PreRun() function will be executed, if defined, for every request. This module is useful to check some user options or to display news about new applications.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'CustomerPanelPreApplicationModule'}->{'CustomerAccept'} = 'Kernel::Modules::CustomerAccept';`

1.42.13. CustomerPanel::InfoKey

Description	Value
Description:	Defines the key to check with CustomerAccept. If this user preferences key is true, then the message is accepted by the system.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'CustomerPanel::InfoKey'} = 'CustomerAccept1';`

1.42.14. CustomerPanel::InfoFile

Description	Value
Description:	Defines the path of the shown info file, that is located under Kernel/Output/HTML/Standard/CustomerAccept.dtl.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'CustomerPanel::InfoFile'} = 'CustomerAccept';`

1.42.15. CustomerPanelLostPassword

Description	Value
Description:	Activates lost password feature for customers.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelLostPassword'} = '1';`

1.42.16. CustomerPanelCreateAccount

Description	Value
Description:	Enables customers to create their own accounts.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelCreateAccount'} = '1';`

1.42.17. CustomerPanelSubjectLostPasswordToken

Description	Value
Description:	Defines the subject for notification mails sent to customers, with token about new requested password.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelSubjectLostPasswordToken'} = 'New OTRS password request';`

1.42.18. CustomerPanelBodyLostPasswordToken

Description	Value
Description:	Defines the body text for notification mails sent to customers, with token about new requested password (after using this link the new password will be sent).
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerPanelBodyLostPasswordToken'} = 'Hi <OTRS_USERFIRSTNAME>,`
	
	`You or someone impersonating you has requested to change your OTRS password.`
	
	`If you want to do this, click on this link. You will receive another email containing the password.`
	
	`<OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/ <OTRS_CONFIG_ScriptAlias>customer.pl? Action=CustomerLostPassword;Token=<OTRS_TOKEN>`
	
	`If you did not request a new password, please ignore this email. ';`

1.42.19. CustomerPanelSubjectLostPassword

Description	Value
Description:	Defines the subject for notification mails sent to customers, about new password.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelSubjectLostPassword'} = 'New OTRS password';`

1.42.20. CustomerPanelBodyLostPassword

Description	Value
Description:	Defines the body text for notification mails sent to customers, about new password (after using this link the new password will be sent).
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerPanelBodyLostPassword'} = 'Hi <OTRS_USERFIRSTNAME>,`
	
	`New password: <OTRS_NEWPW>`
	
	`<OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/`
	`<OTRS_CONFIG_ScriptAlias>customer.pl`
	`';`

1.42.21. CustomerPanelSubjectNewAccount

Description	Value
Description:	Defines the subject for notification mails sent to customers, about new account.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelSubjectNewAccount'} = 'New OTRS Account!';`

1.42.22. CustomerPanelBodyNewAccount

Description	Value
Description:	Defines the body text for notification mails sent to customers, about new account.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerPanelBodyNewAccount'} = 'Hi <OTRS_USERFIRSTNAME>,`
	
	`You or someone impersonating you has created a new OTRS account for you.`
	
	`Full name: <OTRS_USERFIRSTNAME> <OTRS_USERLASTNAME>`
	`User name: <OTRS_USERLOGIN>`
	`Password : <OTRS_USERPASSWORD>`
	
	`You can log in via the following URL. We encourage you to change your password`

Description	Value
	via the Preferences button after logging in. <OTRS_CONFIG_HttpType>://<OTRS_CONFIG_FQDN>/ <OTRS_CONFIG_ScriptAlias>customer.pl ';

1.42.23. Loader::Customer::Skin###000-default

Description	Value
Description:	Default skin for OTRS 3.0 interface.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Loader::Customer::Skin'}->{'000-default'} = { 'Description' => 'This is the default orange - black skin for OTRS 3.0.', 'HomePage' => 'www.otrs.org', 'InternalName' => 'default', 'VisibleName' => 'Default' };`

1.42.24. Loader::Customer::SelectedSkin

Description	Value
Description:	The customer skin's InternalName which should be used in the customer interface. Please check the available skins in Frontend::Customer::Skins.
Group:	Framework
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Loader::Customer::SelectedSkin'} = 'default';`

1.43. Frontend::Customer::Auth

1.43.1. Customer::AuthModule

Description	Value
Description:	Defines the module to authenticate customers.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::DB';`

1.43.2. Customer::AuthModule::DB::CryptType

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, the crypt type of passwords must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Customer::AuthModule::DB::CryptType'} = 'md5';`

1.43.3. Customer::AuthModule::DB::Table

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, the name of the table where your customer data should be stored must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Customer::AuthModule::DB::Table'} = 'customer_user';`

1.43.4. Customer::AuthModule::DB::CustomerKey

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, the name of the column for the CustomerKey in the customer table must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Customer::AuthModule::DB::CustomerKey'} = 'login';`

1.43.5. Customer::AuthModule::DB::CustomerPassword

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, the column name for the CustomerPassword in the customer table must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'Customer::AuthModule::DB::CustomerPassword'} =  'pw';`

1.43.6. Customer::AuthModule::DB::DSN

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, the DSN for the connection to the customer table must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::DB::DSN'} = 'DBI:mysql:database=customerdb;host=customerdbhost';`

1.43.7. Customer::AuthModule::DB::User

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, a username to connect to the customer table can be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::DB::User'} =  'some_user';`

1.43.8. Customer::AuthModule::DB::Password

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, a password to connect to the customer table can be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::DB::Password'} =  'some_password';`

1.43.9. Customer::AuthModule::DB::Type

Description	Value
Description:	If "DB" was selected for Customer::AuthModule, a database driver (normally autodetection is used) can be specified.
Group:	Framework

Description	Value
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::DB::Type'} = 'mysql';`

1.43.10. Customer::AuthModule::HTTPBasicAuth::Replace

Description	Value
Description:	If "HTTPBasicAuth" was selected for Customer::AuthModule, you can specify to strip leading parts of user names (e. g. for domains like example_domain\user to user).
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::HTTPBasicAuth::Replace'} = 'example_domain\\\\';`

1.43.11. Customer::AuthModule::HTTPBasicAuth::ReplaceRegExp

Description	Value
Description:	If "HTTPBasicAuth" was selected for Customer::AuthModule, you can specify (by using a RegExp) to strip parts of REMOTE_USER (e. g. for to remove trailing domains). RegExp-Note, $1 will be the new Login.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::HTTPBasicAuth::ReplaceRegExp'} = '^(.+?)@.+?$';`

1.43.12. Customer::AuthModule::LDAP::Host

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, the LDAP host can be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::Host'} = 'ldap.example.com';`

1.43.13. Customer::AuthModule::LDAP::BaseDN

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, the BaseDN must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';`

1.43.14. Customer::AuthModule::LDAP::UID

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, the user identifier must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::UID'} = 'uid';`

1.43.15. Customer::AuthModule::LDAP::GroupDN

Description	Value
Description:	If "LDAP" was selected for Customer::Authmodule, you can check if the user is allowed to authenticate because he is in a posixGroup, e.g. user needs to be in a group xyz to use OTRS. Specify the group, who may access the system.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::GroupDN'} = 'cn=otrsallow,ou=posixGroups,dc=example,dc=com';`

1.43.16. Customer::AuthModule::LDAP::AccessAttr

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, you can specify access attributes here.
Group:	Framework
SubGroup:	Frontend::Customer::Auth

Description	Value
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::AccessAttr'} = 'memberUid';`

1.43.17. Customer::AuthModule::LDAP::UserAttr

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, user attributes can be specified. For LDAP posixGroups use UID, for non LDAP posixGroups use full user DN.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::UserAttr'} = 'UID';`

1.43.18. Customer::AuthModule::LDAP::SearchUserDN

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule and your users have only anonymous access to the LDAP tree, but you want to search through the data, you can do this with a user who has access to the LDAP directory. Specify the username for this special user here.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::SearchUserDN'} = 'cn=binduser,ou=users,dc=example,dc=com';`

1.43.19. Customer::AuthModule::LDAP::SearchUserPw

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule and your users have only anonymous access to the LDAP tree, but you want to search through the data, you can do this with a user who has access to the LDAP directory. Specify the password for this special user here.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'Customer::AuthModule::LDAP::SearchUserPw'} = 'some_password';`

1.43.20. Customer::AuthModule::LDAP::AlwaysFilter

Description	Value
Description:	If "LDAP" was selected, you can add a filter to each LDAP query, e.g. (mail=*), (objectclass=user) or (!objectclass=computer).
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::AlwaysFilter'} = '(!objectclass=computer)';`

1.43.21. Customer::AuthModule::LDAP::UserSuffix

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule and if you want to add a suffix to every customer login name, specifiy it here, e. g. you just want to write the username user but in your LDAP directory exists user@domain.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::UserSuffix'} = '@domain.com';`

1.43.22. Customer::AuthModule::LDAP::Params

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule and special paramaters are needed for the Net::LDAP perl module, you can specify them here. See "perldoc Net::LDAP" for more information about the parameters.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::Params'} = { 'async' => '0', 'port' => '389', 'timeout' => '120', 'version' => '3' };`

1.43.23. Customer::AuthModule::LDAP::Die

Description	Value
Description:	If "LDAP" was selected for Customer::AuthModule, you can specify if the applications will stop if e. g. a connection to a server can't be established due to network problems.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::LDAP::Die'} = '1';`

1.43.24. Customer::AuthModule::Radius::Host

Description	Value
Description:	If "Radius" was selected for Customer::AuthModule, the radius host must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::Radius::Host'} = 'radiushost';`

1.43.25. Customer::AuthModule::Radius::Password

Description	Value
Description:	If "Radius" was selected for Customer::AuthModule, the password to authenticate to the radius host must be specified.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::Radius::Password'} = 'radiussecret';`

1.43.26. Customer::AuthModule::Radius::Die

Description	Value
Description:	If "Radius" was selected for Customer::AuthModule, you can specify if the applications will stop if e. g. a connection to a server can't be established due to network problems.
Group:	Framework
SubGroup:	Frontend::Customer::Auth
Valid:	1

Description	Value
Required:	0
Config-Setting:	`$Self->{'Customer::AuthModule::Radius::Die'} = '1';`

1.44. Frontend::Customer::ModuleMetaHead

1.44.1. CustomerFrontend::HeaderMetaModule###1-Refresh

Description	Value
Description:	Defines the module to generate html refresh headers of html sites, in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleMetaHead
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerFrontend::HeaderMetaModule'}->{'1-Refresh'} = { 'Module' => 'Kernel::Output::HTML::HeaderMetaRefresh' };`

1.45. Frontend::Customer::ModuleNotify

1.45.1. CustomerFrontend::NotifyModule###1-ShowAgentOnline

Description	Value
Description:	Defines the module that shows the currently loged in agents in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	`$Self->{'CustomerFrontend::NotifyModule'}->{'1-ShowAgentOnline'} = { 'IdleMinutes' => '60', 'Module' => 'Kernel::Output::HTML::NotificationAgentOnline', 'ShowEmail' => '1' };`

1.45.2. CustomerFrontend::NotifyModule###1-ShowCustomerOnline

Description	Value
Description:	Defines the module that shows the currently loged in customers in the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'CustomerFrontend::NotifyModule'}->{'1-ShowCustomerOnline'} = { 'Module' => 'Kernel::Output::HTML::NotificationCustomerOnline', 'ShowEmail' => '1' };`

1.46. Frontend::Customer::ModuleRegistration

1.46.1. CustomerFrontend::Module###Logout

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'Logout'} = { 'Description' => 'Logout of customer panel', 'NavBarName' => '', 'Title' => '' };`

1.46.2. CustomerFrontend::Module###CustomerPreferences

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerPreferences'} = { 'Description' => 'Customer preferences', 'NavBarName' => '', 'Title' => 'Preferences' };`

1.46.3. CustomerFrontend::Module###CustomerAccept

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerAccept'} = { 'Description' => 'To accept login information, such as an EULA or license.',`

Description	Value

	`  'NavBarName' => '',`
	`  'Title' => 'Info'`
	`};`

1.46.4. CustomerFrontend::Module###PictureUpload

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Framework
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	1

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'PictureUpload'} =  {`
	`  'Description' => 'Picture upload module',`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'Picture-Upload'`
	`};`

1.47. Frontend::Customer::Preferences

1.47.1. PreferencesTable

Description	Value
Description:	Defines the name of the table, where the customer preferences are stored.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PreferencesTable'} =  'user_preferences';`

1.47.2. PreferencesTableKey

Description	Value
Description:	Defines the column to store the keys for the preferences table.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PreferencesTableKey'} =  'preferences_key';`

1.47.3. CustomerPreferences

Description	Value
Description:	Defines the parameters for the customer preferences table.

Description	Value
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPreferences'} = { 'Module' => 'Kernel::System::CustomerUser::Preferences::DB', 'Params' => { 'Table' => 'customer_preferences', 'TableKey' => 'preferences_key', 'TableUserID' => 'user_id', 'TableValue' => 'preferences_value' } };`

1.47.4. CustomerPreferencesView

Description	Value
Description:	Sets the order of the different items in the customer preferences view.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPreferencesView'} = [ 'User Profile', 'Other Settings' ];`

1.47.5. CustomerPreferencesGroups###Password

Description	Value
Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerPreferencesGroups'}->{'Password'} = { 'Active' => '1', 'Area' => 'Customer', 'Column' => 'Other Settings', 'Label' => 'Change password', 'Module' => 'Kernel::Output::HTML::PreferencesPassword', 'PasswordMin2Characters' => '0', 'PasswordMin2Lower2UpperCharacters' => '0', 'PasswordMinSize' => '0', 'PasswordNeedDigit' => '0', 'PasswordRegExp' => '', 'Prio' => '1000' };`

1.47.6. CustomerPreferencesGroups###Language

Description	Value
Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'CustomerPreferencesGroups'}->{'Language'} =  {
 'Active' => '1',
 'Column' => 'User Profile',
 'Key' => 'Your language',
 'Label' => 'Interface language',
 'Module' => 'Kernel::Output::HTML::PreferencesLanguage',
 'PrefKey' => 'UserLanguage',
 'Prio' => '2000'
};
```

1.47.7. CustomerPreferencesGroups###Theme

Description	Value
Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'CustomerPreferencesGroups'}->{'Theme'} =  {
 'Active' => '0',
 'Column' => 'User Profile',
 'Key' => 'Select your frontend Theme.',
 'Label' => 'Theme',
 'Module' => 'Kernel::Output::HTML::PreferencesTheme',
 'PrefKey' => 'UserTheme',
 'Prio' => '1000'
};
```

1.47.8. CustomerPreferencesGroups###TimeZone

Description	Value
Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'CustomerPreferencesGroups'}->{'TimeZone'} =  {
 'Active' => '1',
 'Column' => 'User Profile',
 'Key' => 'Time Zone',
 'Label' => 'Time Zone',
```

Description	Value

	`'Module' => 'Kernel::Output::HTML::PreferencesTimeZone',`
	`'PrefKey' => 'UserTimeZone',`
	`'Prio' => '5000'`
	`};`

1.47.9. CustomerPreferencesGroups###PGP

Description	Value
Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerPreferencesGroups'}->{'PGP'} = {`
	`'Active' => '1',`
	`'Column' => 'Other Settings',`
	`'Key' => 'PGP Key Upload',`
	`'Label' => 'PGP Key',`
	`'Module' => 'Kernel::Output::HTML::PreferencesPGP',`
	`'PrefKey' => 'UserPGPKey',`
	`'Prio' => '10000'`
	`};`

1.47.10. CustomerPreferencesGroups###SMIME

Description	Value
Description:	Defines all the parameters for this item in the customer preferences.
Group:	Framework
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'CustomerPreferencesGroups'}->{'SMIME'} = {`
	`'Active' => '1',`
	`'Column' => 'Other Settings',`
	`'Key' => 'S/MIME Certificate Upload',`
	`'Label' => 'S/MIME Certificate',`
	`'Module' => 'Kernel::Output::HTML::PreferencesSMIME',`
	`'PrefKey' => 'UserSMIMEKey',`
	`'Prio' => '11000'`
	`};`

1.48. Frontend::Public

1.48.1. PublicFrontend::CommonParam###Action

Description	Value
Description:	Defines the default value for the action parameter for the public frontend. The action parameter is used in the scripts of the system.
Group:	Framework
SubGroup:	Frontend::Public

Description	Value
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PublicFrontend::CommonParam'}->{'Action'} = 'PublicDefault';`

1.49. Frontend::Public::ModuleRegistration

1.49.1. PublicFrontend::Module###PublicDefault

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Public::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'PublicFrontend::Module'}->{'PublicDefault'} = { 'Description' => 'PublicDefault', 'NavBarName' => '', 'Title' => 'PublicDefault' };`

1.49.2. PublicFrontend::Module###PublicRepository

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Framework
SubGroup:	Frontend::Public::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'PublicFrontend::Module'}->{'PublicRepository'} = { 'Description' => 'PublicRepository', 'NavBarName' => '', 'Title' => 'PublicRepository' };`

2. Ticket

2.1. Core

2.1.1. OTRSEscalationEvents::DecayTime

Description	Value
Description:	The duration in minutes after emitting an event, in which the new escalation notify and start events are suppressed.
Group:	Ticket
SubGroup:	Core
Valid:	1

Description	Value
Required:	0
Config-Setting:	`$Self->{'OTRSEscalationEvents::DecayTime'} = '1440';`

2.2. Core::FulltextSearch

2.2.1. Ticket::SearchIndexModule

Description	Value
Description:	Helps to extend your articles full-text search (From, To, Cc, Subject and Body search). Runtime will do full-text searches on live data (it works fine for up to 50.000 tickets). StaticDB will strip all articles and will build an index after article creation, increasing fulltext searches about 50%. To create an initial index use "bin/otrs.RebuildFulltextIndex.pl".
Group:	Ticket
SubGroup:	Core::FulltextSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::SearchIndexModule'} = 'Kernel::System::Ticket::ArticleSearchIndex::RuntimeDB';`

2.2.2. Ticket::SearchIndex::Attribute

Description	Value
Description:	Configures the full-text index. Execute "bin/otrs.RebuildFulltextIndex.pl" in order to generate a new index.
Group:	Ticket
SubGroup:	Core::FulltextSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::SearchIndex::Attribute'} = { 'WordCountMax' => '1000', 'WordLengthMax' => '30', 'WordLengthMin' => '3' };`

2.2.3. Ticket::EventModulePost###98-ArticleSearchIndex

Description	Value
Description:	Builds an article index right after the article's creation.
Group:	Ticket
SubGroup:	Core::FulltextSearch
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'Ticket::EventModulePost'}->{'98-ArticleSearchIndex'} = { 'Event' => '(ArticleCreate|ArticleUpdate)', 'Module' => 'Kernel::System::Ticket::Event::ArticleSearchIndex' };`

2.3. Core::LinkObject

2.3.1. LinkObject::PossibleLink###0200

Description	Value
Description:	Links 2 tickets with a "Normal" type link.
Group:	Ticket
SubGroup:	Core::LinkObject
Valid:	1
Required:	0
Config-Setting:	`$Self->{'LinkObject::PossibleLink'}->{'0200'} = { 'Object1' => 'Ticket', 'Object2' => 'Ticket', 'Type' => 'Normal' };`

2.3.2. LinkObject::PossibleLink###0201

Description	Value
Description:	Links 2 tickets with a "ParentChild" type link.
Group:	Ticket
SubGroup:	Core::LinkObject
Valid:	1
Required:	0
Config-Setting:	`$Self->{'LinkObject::PossibleLink'}->{'0201'} = { 'Object1' => 'Ticket', 'Object2' => 'Ticket', 'Type' => 'ParentChild' };`

2.4. Core::PostMaster

2.4.1. PostmasterMaxEmails

Description	Value
Description:	Maximal auto email responses to own email-address a day (Loop-Protection).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterMaxEmails'} = '40';`

2.4.2. PostMasterMaxEmailSize

Description	Value
Description:	Maximal size in KBytes for mails that can be fetched via POP3/POP3S/IMAP/IMAPS (KBytes).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostMasterMaxEmailSize'} =  '16384';`

2.4.3. PostMasterReconnectMessage

Description	Value
Description:	The "bin/PostMasterMailAccount.pl" will reconnect to POP3/POP3S/IMAP/IMAPS host after the specified count of messages.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostMasterReconnectMessage'} =  '20';`

2.4.4. LoopProtectionModule

Description	Value
Description:	Default loop protection module.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'LoopProtectionModule'} = 'Kernel::System::PostMaster::LoopProtection::DB';`

2.4.5. LoopProtectionLog

Description	Value
Description:	Path for the log file (it only applies if "FS" was selected for LoopProtectionModule and it is mandatory).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'LoopProtectionLog'} = '<OTRS_CONFIG_Home>/var/log/LoopProtection';`

2.4.6. PostmasterAutoHTML2Text

Description	Value
Description:	Converts HTML mails into text messages.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterAutoHTML2Text'} = '1';`

2.4.7. PostmasterFollowUpSearchInReferences

Description	Value
Description:	Executes follow up checks on In-Reply-To or References headers for mails that don't have a ticket number in the subject.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpSearchInReferences'} = '0';`

2.4.8. PostmasterFollowUpSearchInBody

Description	Value
Description:	Executes follow up mail body checks in mails that don't have a ticket number in the subject.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpSearchInBody'} = '0';`

2.4.9. PostmasterFollowUpSearchInAttachment

Description	Value
Description:	Executes follow up mail attachments checks in mails that don't have a ticket number in the subject.
Group:	Ticket
SubGroup:	Core::PostMaster

Description	Value
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpSearchInAttachment'} = '0';`

2.4.10. PostmasterFollowUpSearchInRaw

Description	Value
Description:	Executes follow up plain/raw mail checks in mails that don't have a ticket number in the subject.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpSearchInRaw'} = '0';`

2.4.11. PostmasterUserID

Description	Value
Description:	Specifies user id of the postmaster data base.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterUserID'} = '1';`

2.4.12. PostmasterDefaultQueue

Description	Value
Description:	Defines the postmaster default queue.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterDefaultQueue'} = 'Raw';`

2.4.13. PostmasterDefaultPriority

Description	Value
Description:	Defines the default priority of new tickets.
Group:	Ticket
SubGroup:	Core::PostMaster

Description	Value
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterDefaultPriority'} = '3 normal';`

2.4.14. PostmasterDefaultState

Description	Value
Description:	Defines the default state of new tickets.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterDefaultState'} = 'new';`

2.4.15. PostmasterFollowUpState

Description	Value
Description:	Defines the state of a ticket if it gets a follow-up.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpState'} = 'open';`

2.4.16. PostmasterFollowUpStateClosed

Description	Value
Description:	Defines the state of a ticket if it gets a follow-up and the ticket was already closed.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	`$Self->{'PostmasterFollowUpStateClosed'} = 'open';`

2.4.17. PostmasterFollowUpOnUnlockAgentNotifyOnlyToOwner

Description	Value
Description:	Sends agent follow-up notification only to the owner, if a ticket is unlocked (the default is to send the notification to all agents).
Group:	Ticket

Description	Value
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostmasterFollowUpOnUnlockAgentNotifyOnlyToOwner'} = '0';`

2.4.18. PostmasterX-Header

Description	Value
Description:	Defines all the X-headers that should be scanned.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'PostmasterX-Header'} = [
 'From',
 'To',
 'Cc',
 'Reply-To',
 'ReplyTo',
 'Subject',
 'Message-ID',
 'Message-Id',
 'Resent-To',
 'Resent-From',
 'Precedence',
 'Mailing-List',
 'List-Id',
 'List-Archive',
 'Errors-To',
 'References',
 'In-Reply-To',
 'X-Loop',
 'X-Spam-Flag',
 'X-Spam-Level',
 'X-Spam-Score',
 'X-Spam-Status',
 'X-No-Loop',
 'X-Priority',
 'Importance',
 'X-Mailer',
 'User-Agent',
 'Organization',
 'X-Original-To',
 'Delivered-To',
 'Envelope-To',
 'Return-Path',
 'X-OTRS-Loop',
 'X-OTRS-Info',
 'X-OTRS-Priority',
 'X-OTRS-Queue',
 'X-OTRS-Lock',
 'X-OTRS-Ignore',
 'X-OTRS-State',
 'X-OTRS-State-PendingTime',
 'X-OTRS-Type',
 'X-OTRS-Service',
```

Description	Value

	`  'X-OTRS-SLA',`
	`  'X-OTRS-CustomerNo',`
	`  'X-OTRS-CustomerUser',`
	`  'X-OTRS-SenderType',`
	`  'X-OTRS-ArticleType',`
	`  'X-OTRS-FollowUp-Priority',`
	`  'X-OTRS-FollowUp-Queue',`
	`  'X-OTRS-FollowUp-Lock',`
	`  'X-OTRS-FollowUp-State',`
	`  'X-OTRS-FollowUp-State-PendingTime',`
	`  'X-OTRS-FollowUp-Type',`
	`  'X-OTRS-FollowUp-Service',`
	`  'X-OTRS-FollowUp-SLA',`
	`  'X-OTRS-FollowUp-SenderType',`
	`  'X-OTRS-FollowUp-ArticleType'`
	`];`

2.4.19. PostMaster::PreFilterModule###1-Match

Description	Value
Description:	Module to filter and manipulate incoming messages. Block/ignore all spam email with From: noreply@ address.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	`$Self->{'PostMaster::PreFilterModule'}->{'1-Match'} =  {`
	`  'Match' => {`
	`  'From' => 'noreply@'`
	`  },`
	`  'Module' => 'Kernel::System::PostMaster::Filter::Match',`
	`  'Set' => {`
	`  'X-OTRS-Ignore' => 'yes'`
	`  },`
	`  'StopAfterMatch' => '0'`
	`};`

2.4.20. PostMaster::PreFilterModule###2-Match

Description	Value
Description:	Module to filter and manipulate incoming messages. Get a 4 digit number to ticket free text, use regex in Match e. g. From => '(.+?)@.+?', and use () as [***] in Set =>.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	`$Self->{'PostMaster::PreFilterModule'}->{'2-Match'} =  {`
	`  'Match' => {`
	`  'Subject' => 'SomeNumber:(\\d\\d\\d\\d)'`
	`  },`
	`  'Module' => 'Kernel::System::PostMaster::Filter::Match',`
	`  'Set' => {`

Description	Value

	`'X-OTRS-DynamicField-TicketFreeKey1' => 'SomeNumber',`
	`'X-OTRS-DynamicField-TicketFreeText1' => '[***]'`
	`},`
	`'StopAfterMatch' => '0'`
	`};`

2.4.21. PostMaster::PreFilterModule###3-NewTicketReject

Description	Value
Description:	Blocks all the incoming emails that do not have a valid ticket number in subject with From: @example.com address.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	`$Self->{'PostMaster::PreFilterModule'}->{'3-NewTicketReject'} = {`
	`'Match' => {`
	`'From' => '@example.com'`
	`},`
	`'Module' => 'Kernel::System::PostMaster::Filter::NewTicketReject',`
	`'Set' => {`
	`'X-OTRS-Ignore' => 'yes'`
	`},`
	`'StopAfterMatch' => '0'`
	`};`

2.4.22. PostMaster::PreFilterModule::NewTicketReject::Sender

Description	Value
Description:	Defines the sender for rejected emails.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	`$Self->{'PostMaster::PreFilterModule::NewTicketReject::Sender'} =` `'noreply@example.com';`

2.4.23. PostMaster::PreFilterModule::NewTicketReject::Subject

Description	Value
Description:	Defines the subject for rejected emails.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostMaster::PreFilterModule::NewTicketReject::Subject'} =` `'Email Rejected';`

2.4.24. PostMaster::PreFilterModule::NewTicketReject::Body

Description	Value
Description:	Defines the body text for rejected emails.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'PostMaster::PreFilterModule::NewTicketReject::Body'} = ' Dear Customer, Unfortunately we could not detect a valid ticket number in your subject, so this email can\'t be processed. Please create a new ticket via the customer panel. Thanks for your help! Your Helpdesk Team ';`

2.4.25. PostMaster::PreFilterModule###4-CMD

Description	Value
Description:	CMD example setup. Ignores emails where external CMD returns some output on STDOUT (email will be piped into STDIN of some.bin).
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	`$Self->{'PostMaster::PreFilterModule'}->{'4-CMD'} = { 'CMD' => '/usr/bin/some.bin', 'Module' => 'Kernel::System::PostMaster::Filter::CMD', 'Set' => { 'X-OTRS-Ignore' => 'yes' } };`

2.4.26. PostMaster::PreFilterModule###5-SpamAssassin

Description	Value
Description:	Spam Assassin example setup. Ignores emails that are marked with SpamAssassin.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0
Config-Setting:	`$Self->{'PostMaster::PreFilterModule'}->{'5-SpamAssassin'} = { 'CMD' => '/usr/bin/spamassassin | grep -i "X-Spam-Status: yes"',`

Description	Value

	`'Module' => 'Kernel::System::PostMaster::Filter::CMD',`
	`'Set' => {`
	`'X-OTRS-Ignore' => 'yes'`
	`}`
	`};`

2.4.27. PostMaster::PreFilterModule###6-SpamAssassin

Description	Value
Description:	Spam Assassin example setup. Moves marked mails to spam queue.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	0
Required:	0

Config-Setting:	`$Self->{'PostMaster::PreFilterModule'}->{'6-SpamAssassin'} = {`
	`'CMD' => '/usr/bin/spamassassin | grep -i "X-Spam-Status: yes"',`
	`'Module' => 'Kernel::System::PostMaster::Filter::CMD',`
	`'Set' => {`
	`'X-OTRS-Queue' => 'spam'`
	`}`
	`};`

2.4.28. PostMaster::PreFilterModule###000-MatchDBSource

Description	Value
Description:	Module to use database filter storage.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1

Config-Setting:	`$Self->{'PostMaster::PreFilterModule'}->{'000-MatchDBSource'} = {`
	`'Module' => 'Kernel::System::PostMaster::Filter::MatchDBSource'`
	`};`

2.4.29. PostMaster::PostFilterModule###000-FollowUpArticleTypeCheck

Description	Value
Description:	Module to check if arrived emails should be marked as email-internal (because of original forwared internal email it college). ArticleType and SenderType define the values for the arrived email/article.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	0
Config-Setting:	`$Self->{'PostMaster::PostFilterModule'}->{'000-FollowUpArticleTypeCheck'} = {` `'ArticleType' => 'email-internal',`

Description	Value
	`  'Module' => 'Kernel::System::PostMaster::Filter::FollowUpArticleTypeCheck', 'SenderType' => 'customer' };`

2.4.30. SendNoAutoResponseRegExp

Description	Value
Description:	If this regex matches, no message will be send by the autoresponder.
Group:	Ticket
SubGroup:	Core::PostMaster
Valid:	1
Required:	1
Config-Setting:	`$Self->{'SendNoAutoResponseRegExp'} = '(MAILER-DAEMON|postmaster|abuse)@.+?\\..+?';`

2.5. Core::Stats

2.5.1. Stats::DynamicObjectRegistration###Ticket

Description	Value
Description:	Module to generate ticket statistics.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Stats::DynamicObjectRegistration'}->{'Ticket'} = { 'Module' => 'Kernel::System::Stats::Dynamic::Ticket' };`

2.5.2. Stats::DynamicObjectRegistration###TicketList

Description	Value
Description:	Determines if the statistics module may generate ticket lists.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Stats::DynamicObjectRegistration'}->{'TicketList'} = { 'Module' => 'Kernel::System::Stats::Dynamic::TicketList' };`

2.5.3. Stats::DynamicObjectRegistration###TicketAccountedTime

Description	Value
Description:	Module to generate accounted time ticket statistics.

Description	Value
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Stats::DynamicObjectRegistration'}->{'TicketAccountedTime'} = { 'Module' => 'Kernel::System::Stats::Dynamic::TicketAccountedTime' };`

2.5.4. Stats::DynamicObjectRegistration###TicketSolutionResponseTime

Description	Value
Description:	Module to generate ticket solution and response time statistics.
Group:	Ticket
SubGroup:	Core::Stats
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Stats::DynamicObjectRegistration'}->{'TicketSolutionResponseTime'} = { 'Module' => 'Kernel::System::Stats::Dynamic::TicketSolutionResponseTime' };`

2.6. Core::Ticket

2.6.1. Ticket::Hook

Description	Value
Description:	The identifier for a ticket, e.g. Ticket#, Call#, MyTicket#. The default is Ticket#.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Hook'} = 'Ticket#';`

2.6.2. Ticket::HookDivider

Description	Value
Description:	The divider between TicketHook and ticket number. E.g ': '.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Description	Value
Config-Setting:	`$Self->{'Ticket::HookDivider'} = '';`

2.6.3. Ticket::SubjectSize

Description	Value
Description:	Max size of the subjects in an email reply.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::SubjectSize'} = '100';`

2.6.4. Ticket::SubjectRe

Description	Value
Description:	The text at the beginning of the subject in an email reply, e.g. RE, AW, or AS.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::SubjectRe'} = 'Re';`

2.6.5. Ticket::SubjectFwd

Description	Value
Description:	The text at the beginning of the subject when an email is forwarded, e.g. FW, Fwd, or WG.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::SubjectFwd'} = 'Fwd';`

2.6.6. Ticket::SubjectFormat

Description	Value
Description:	The format of the subject. 'Left' means '[TicketHook#:12345] Some Subject', 'Right' means 'Some Subject [TicketHook#:12345]', 'None' means 'Some Subject' and no ticket number. In the last case you should enable PostmasterFollowupSearchInRaw or PostmasterFollowUpSearchInReferences to recognize followups based on email headers and/or body.

Description	Value
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::SubjectFormat'} = 'Left';`

2.6.7. Ticket::CustomQueue

Description	Value
Description:	Name of custom queue. The custom queue is a queue selection of your preferred queues and can be selected in the preferences settings.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomQueue'} = 'My Queues';`

2.6.8. Ticket::NewArticleIgnoreSystemSender

Description	Value
Description:	Ignore article with system sender type for new article feature (e. g. auto responses or email notifications).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::NewArticleIgnoreSystemSender'} = '0';`

2.6.9. Ticket::ChangeOwnerToEveryone

Description	Value
Description:	Changes the owner of tickets to everyone (useful for ASP). Normally only agent with rw permissions in the queue of the ticket will be shown.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ChangeOwnerToEveryone'} = '0';`

2.6.10. Ticket::Responsible

Description	Value
Description:	Enables ticket responsible feature, to keep track of a specific ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Responsible'} =  '0';`

2.6.11. Ticket::ResponsibleAutoSet

Description	Value
Description:	Automatically sets the owner of a ticket as the responsible for it (if ticket responsible feature is enabled).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::ResponsibleAutoSet'} =  '1';`

2.6.12. Ticket::Type

Description	Value
Description:	Allows defining new types for ticket (if ticket type feature is enabled).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Type'} =  '0';`

2.6.13. Ticket::Service

Description	Value
Description:	Allows defining services and SLAs for tickets (e. g. email, desktop, network, ...), and escalation attributes for SLAs (if ticket service/SLA feature is enabled).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'Ticket::Service'} = '0';`

2.6.14. Ticket::Service::Default::UnknownCustomer

Description	Value
Description:	Allows default services to be selected also for non existing customers.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Service::Default::UnknownCustomer'} = '0';`

2.6.15. Ticket::ArchiveSystem

Description	Value
Description:	Activates the ticket archive system to have a faster system by moving some tickets out of the daily scope. To search for these tickets, the archive flag has to be enabled in the ticket search.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ArchiveSystem'} = '0';`

2.6.16. Ticket::CustomerArchiveSystem

Description	Value
Description:	Activates the ticket archive system search in the customer interface.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomerArchiveSystem'} = '0';`

2.6.17. Ticket::NumberGenerator

Description	Value
Description:	Selects the ticket number generator module. "AutoIncrement" increments the ticket number, the SystemID and the counter are used with SystemID.counter format (e.g. 1010138, 1010139). With "Date" the ticket numbers will be generated by the current date, the SystemID and the counter. The format looks

Description	Value
	like Year.Month.Day.SystemID.counter (e.g. 200206231010138, 200206231010139). With "DateChecksum" the counter will be appended as checksum to the string of date and SystemID. The checksum will be rotated on a daily basis. The format looks like Year.Month.Day.SystemID.Counter.CheckSum (e.g. 2002070110101520, 2002070110101535). "Random" generates randomized ticket numbers in the format "SystemID.Random" (e.g. 100057866352, 103745394596).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::NumberGenerator'} = 'Kernel::System::Ticket::Number::DateChecksum';`

2.6.18. Ticket::NumberGenerator::MinCounterSize

Description	Value
Description:	Sets the minimal ticket counter size (if "AutoIncrement" was selected as TicketNumberGenerator). Default is 5, this means the counter starts from 10000.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::NumberGenerator::MinCounterSize'} = '5';`

2.6.19. Ticket::NumberGenerator::CheckSystemID

Description	Value
Description:	Checks the SystemID in ticket number detection for follow-ups (use "No" if SystemID has been changed after using the system).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::NumberGenerator::CheckSystemID'} = '1';`

2.6.20. Ticket::CounterLog

Description	Value
Description:	Log file for the ticket counter.
Group:	Ticket

Description	Value
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CounterLog'} = '<OTRS_CONFIG_Home>/var/log/TicketCounter.log';`

2.6.21. Ticket::IndexModule

Description	Value
Description:	IndexAccelerator: to choose your backend TicketViewAccelerator module. "RuntimeDB" generates each queue view on the fly from ticket table (no performance problems up to approx. 60.000 tickets in total and 6.000 open tickets in the system). "StaticDB" is the most powerful module, it uses an extra ticket-index table that works like a view (recommended if more than 80.000 and 6.000 open tickets are stored in the system). Use the script "bin/otrs.RebuildTicketIndex.pl" for initial index update.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::IndexModule'} = 'Kernel::System::Ticket::IndexAccelerator::RuntimeDB';`

2.6.22. Ticket::StorageModule

Description	Value
Description:	Saves the attachments of articles. "DB" stores all data in the database (not recommended for storing big attachments). "FS" stores the data on the filesystem; this is faster but the webserver should run under the OTRS user. You can switch between the modules even on a system that is already in production without any loss of data.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::StorageModule'} = 'Kernel::System::Ticket::ArticleStorageDB';`

2.6.23. ArticleDir

Description	Value
Description:	Specifies the directory to store the data in, if "FS" was selected for TicketStorageModule.

Description	Value
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'ArticleDir'} =  '<OTRS_CONFIG_Home>/var/article';`

2.6.24. Ticket::EventModulePost###100-ArchiveRestore

Description	Value
Description:	Restores a ticket from the archive (only if the event is a state change, from closed to any open available state).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'100-ArchiveRestore'} =  { 'Event' => 'TicketStateUpdate', 'Module' => 'Kernel::System::Ticket::Event::ArchiveRestore' };`

2.6.25. Ticket::EventModulePost###110-AcceleratorUpdate

Description	Value
Description:	Updates the ticket index accelerator.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'110-AcceleratorUpdate'} =  { 'Event' => 'TicketStateUpdate|TicketQueueUpdate|TicketLockUpdate', 'Module' => 'Kernel::System::Ticket::Event::TicketAcceleratorUpdate' };`

2.6.26. Ticket::EventModulePost###120-ForceOwnerResetOnMove

Description	Value
Description:	Resets and unlocks the owner of a ticket if it was moved to another queue.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'Ticket::EventModulePost'}->{'120-ForceOwnerResetOnMove'} = { 'Event' => 'TicketQueueUpdate', 'Module' => 'Kernel::System::Ticket::Event::ForceOwnerReset' };`

2.6.27. Ticket::EventModulePost###130-ForceStateChangeOnLock

Description	Value
Description:	Forces to choose a different ticket state (from current) after lock action. Define the current state as key, and the next state after lock action as content.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'130-ForceStateChangeOnLock'} = { 'Event' => 'TicketLockUpdate', 'Module' => 'Kernel::System::Ticket::Event::ForceState', 'new' => 'open' };`

2.6.28. Ticket::EventModulePost###140-ResponsibleAutoSet

Description	Value
Description:	Automatically sets the responsible of a ticket (if it is not set yet) after the first owner update.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'140-ResponsibleAutoSet'} = { 'Event' => 'TicketOwnerUpdate', 'Module' => 'Kernel::System::Ticket::Event::ResponsibleAutoSet' };`

2.6.29. Ticket::EventModulePost###150-TicketPendingTimeReset

Description	Value
Description:	Sets the PendingTime of a ticket to 0 if the state is changed to a non-pending state.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Description	Value
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'150-TicketPendingTimeReset'} = { 'Event' => 'TicketStateUpdate', 'Module' => 'Kernel::System::Ticket::Event::TicketPendingTimeReset' };`

2.6.30. Ticket::EventModulePost###500-NotificationEvent

Description	Value
Description:	Sends the notifications which are configured in the admin interface under "Notfication (Event)".
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'500-NotificationEvent'} = { 'Event' => '', 'Module' => 'Kernel::System::Ticket::Event::NotificationEvent', 'Transaction' => '1' };`

2.6.31. Ticket::EventModulePost###900-EscalationIndex

Description	Value
Description:	Updates the ticket escalation index after a ticket attribute got updated.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'900-EscalationIndex'} = { 'Event' => 'TicketSLAUpdate|TicketQueueUpdate|TicketStateUpdate|TicketCreate|ArticleCreate', 'Module' => 'Kernel::System::Ticket::Event::TicketEscalationIndex' };`

2.6.32. Ticket::EventModulePost###900-EscalationStopEvents

Description	Value
Description:	Ticket event module that triggers the escalation stop events.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'Ticket::EventModulePost'}->{'900-EscalationStopEvents'} = { 'Event' => 'TicketSLAUpdate|TicketQueueUpdate|TicketStateUpdate|ArticleCreate', 'Module' => 'Kernel::System::Ticket::Event::TriggerEscalationStopEvents' };`

2.6.33. Ticket::EventModulePost###910-ForceUnlockOnMove

Description	Value
Description:	Forces to unlock tickets after being moved to another queue.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'910-ForceUnlockOnMove'} = { 'Event' => 'TicketQueueUpdate', 'Module' => 'Kernel::System::Ticket::Event::ForceUnlock' };`

2.6.34. Ticket::EventModulePost###920-TicketArticleNewMessageUpdate

Description	Value
Description:	Update Ticket "Seen" flag if every article got seen or a new Article got created.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::EventModulePost'}->{'920-TicketArticleNewMessageUpdate'} = { 'Event' => 'ArticleCreate|ArticleFlagSet', 'Module' => 'Kernel::System::Ticket::Event::TicketNewMessageUpdate' };`

2.6.35. Ticket::CustomModule###001-CustomModule

Description	Value
Description:	Overloads (redefines) existing functions in Kernel::System::Ticket. Used to easily add customizations.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::CustomModule'}->{'001-CustomModule'} = 'Kernel::System::Ticket::Custom';`

2.6.36. Ticket::ViewableSenderTypes

Description	Value
Description:	Defines the default viewable sender types of a ticket (default: customer).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ViewableSenderTypes'} = [ '\'customer\'' ];`

2.6.37. Ticket::ViewableLocks

Description	Value
Description:	Defines the viewable locks of a ticket. Default: unlock, tmp_lock.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ViewableLocks'} = [ '\'unlock\'', '\'tmp_lock\'' ];`

2.6.38. Ticket::ViewableStateType

Description	Value
Description:	Defines the valid state types for a ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ViewableStateType'} = [ 'new', 'open', 'pending reminder', 'pending auto' ];`

2.6.39. Ticket::UnlockStateType

Description	Value
Description:	Defines the valid states for unlocked tickets. To unlock tickets the script "bin/otrs.UnlockTickets.pl" can be used.

Description	Value
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::UnlockStateType'} = [ 'new', 'open' ];`

2.6.40. Ticket::PendingNotificationOnlyToOwner

Description	Value
Description:	Sends reminder notifications of unlocked ticket after reaching the reminder date (only sent to ticket owner).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::PendingNotificationOnlyToOwner'} = '0';`

2.6.41. Ticket::PendingNotificationNotToResponsible

Description	Value
Description:	Disables sending reminder notifications to the responsible agent of a ticket (Ticket::Responsible needs to be activated).
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::PendingNotificationNotToResponsible'} = '0';`

2.6.42. Ticket::PendingReminderStateType

Description	Value
Description:	Defines the state type of the reminder for pending tickets.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::PendingReminderStateType'} = [ 'pending reminder'`

Description	Value
	`];`

2.6.43. Ticket::PendingAutoStateType

Description	Value
Description:	Determines the possible states for pending tickets that changed state after reaching time limit.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::PendingAutoStateType'} = [`
	`  'pending auto'`
	`];`

2.6.44. Ticket::StateAfterPending

Description	Value
Description:	Defines which states should be set automatically (Content), after the pending time of state (Key) has been reached.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::StateAfterPending'} = {`
	`  'pending auto close+' => 'closed successful',`
	`  'pending auto close-' => 'closed unsuccessful'`
	`};`

2.6.45. System::Permission

Description	Value
Description:	Standard available permissions for agents within the application. If more permissions are needed, they can be entered here. Permissions must be defined to be effective. Some other good permissions have also been provided built-in: note, close, pending, customer, freetext, move, compose, responsible, forward, and bounce. Make sure that "rw" is always the last registered permission.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1

Config-Setting:	`$Self->{'System::Permission'} = [`
	`  'ro',`
	`  'move_into',`
	`  'create',`

Description	Value

	`'note',`
	`'owner',`
	`'priority',`
	`'rw'`
	`];`

2.6.46. Ticket::Permission###1-OwnerCheck

Description	Value
Description:	Module to check the owner of a ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Permission'}->{'1-OwnerCheck'} = {`
	`'Granted' => '1',`
	`'Module' => 'Kernel::System::Ticket::Permission::OwnerCheck',`
	`'Required' => '0'`
	`};`

2.6.47. Ticket::Permission###2-ResponsibleCheck

Description	Value
Description:	Module to check the agent responsible of a ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Permission'}->{'2-ResponsibleCheck'} = {`
	`'Granted' => '1',`
	`'Module' => 'Kernel::System::Ticket::Permission::ResponsibleCheck',`
	`'Required' => '0'`
	`};`

2.6.48. Ticket::Permission###3-GroupCheck

Description	Value
Description:	Module to check if a user is in a special group. Access is granted, if the user is in the specified group and has ro and rw permissions.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Permission'}->{'3-GroupCheck'} = {`
	`'Granted' => '1',`
	`'Module' => 'Kernel::System::Ticket::Permission::GroupCheck',`
	`'Required' => '0'`

Description	Value
	`};`

2.6.49. Ticket::Permission###4-WatcherCheck

Description	Value
Description:	Module to check the watcher agents of a ticket.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Permission'}->{'4-WatcherCheck'} = { 'Granted' => '1', 'Module' => 'Kernel::System::Ticket::Permission::WatcherCheck', 'Required' => '0' };`

2.6.50. CustomerTicket::Permission###1-GroupCheck

Description	Value
Description:	Module to check the group permissions for the access to customer tickets.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerTicket::Permission'}->{'1-GroupCheck'} = { 'Granted' => '0', 'Module' => 'Kernel::System::Ticket::CustomerPermission::GroupCheck', 'Required' => '1' };`

2.6.51. CustomerTicket::Permission###2-CustomerUserIDCheck

Description	Value
Description:	Grants access, if the customer ID of the ticket matches the customer user's ID and the customer user has group permissions on the queue the ticket is in.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerTicket::Permission'}->{'2-CustomerUserIDCheck'} = { 'Granted' => '1', 'Module' => 'Kernel::System::Ticket::CustomerPermission::CustomerUserIDCheck', 'Required' => '0'`

Description	Value
	`};`

2.6.52. CustomerTicket::Permission###3-CustomerIDCheck

Description	Value
Description:	Module to check customer permissions.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerTicket::Permission'}->{'3-CustomerIDCheck'} = { 'Granted' => '1', 'Module' => 'Kernel::System::Ticket::CustomerPermission::CustomerIDCheck', 'Required' => '0' };`

2.6.53. Ticket::DefineEmailFrom

Description	Value
Description:	Defines how the From field from the emails (sent from answers and email tickets) should look like.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::DefineEmailFrom'} = 'SystemAddressName';`

2.6.54. Ticket::DefineEmailFromSeparator

Description	Value
Description:	Defines the separator between the agents real name and the given queue email address.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::DefineEmailFromSeparator'} = 'via';`

2.6.55. CustomerNotifyJustToRealCustomer

Description	Value
Description:	Sends customer notifications just to the mapped customer. Normally, if no customer is mapped, the latest customer sender gets the notification.

Description	Value
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerNotifyJustToRealCustomer'} = '0';`

2.6.56. AgentSelfNotifyOnAction

Description	Value
Description:	Specifies if an agent should receive email notification of his own actions.
Group:	Ticket
SubGroup:	Core::Ticket
Valid:	1
Required:	1
Config-Setting:	`$Self->{'AgentSelfNotifyOnAction'} = '0';`

2.7. Core::TicketACL

2.7.1. Ticket::Acl::Module###1-Ticket::Acl::Module

Description	Value
Description:	ACL module that allows closing parent tickets only if all its children are already closed ("State" shows which states are not available for the parent ticket until all child tickets are closed).
Group:	Ticket
SubGroup:	Core::TicketACL
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Acl::Module'}->{'1-Ticket::Acl::Module'} = { 'Module' => 'Kernel::System::Ticket::Acl::CloseParentAfterClosedChilds', 'State' => [ 'closed successful', 'closed unsuccessful' ] };`

2.7.2. TicketACL::Default::Action

Description	Value
Description:	Default ACL values for ticket actions.
Group:	Ticket
SubGroup:	Core::TicketACL

Description	Value
Valid:	1
Required:	1
Config-Setting:	`$Self->{'TicketACL::Default::Action'} = {};`

2.8. Core::TicketBulkAction

2.8.1. Ticket::Frontend::BulkFeature

Description	Value
Description:	Enables ticket bulk action feature for the agent frontend to work on more than one ticket at a time.
Group:	Ticket
SubGroup:	Core::TicketBulkAction
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::BulkFeature'} = '1';`

2.8.2. Ticket::Frontend::BulkFeatureGroup

Description	Value
Description:	Enables ticket bulk action feature only for the listed groups.
Group:	Ticket
SubGroup:	Core::TicketBulkAction
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::BulkFeatureGroup'} = [ 'admin', 'users' ];`

2.9. Core::TicketDynamicFieldDefault

2.9.1. Ticket::EventModulePost###TicketDynamicFieldDefault

Description	Value
Description:	Event module registration. For more performance you can define a trigger event (e. g. Event => TicketCreate). This is only possible if all Ticket dynamic fields need the same event.
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'Ticket::EventModulePost'}->{'TicketDynamicFieldDefault'} = { 'Module' => 'Kernel::System::Ticket::Event::TicketDynamicFieldDefault', 'Transaction' => '1' };`

2.9.2. Ticket::TicketDynamicFieldDefault###Element1

Description	Value
Description:	Configures a default TicketDynmicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element1'} = { 'Event' => 'TicketCreate', 'Name' => 'Field1', 'Value' => 'Default' };`

2.9.3. Ticket::TicketDynamicFieldDefault###Element2

Description	Value
Description:	Configures a default TicketDynmicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element2'} = { 'Event' => '', 'Name' => '', 'Value' => '' };`

2.9.4. Ticket::TicketDynamicFieldDefault###Element3

Description	Value
Description:	Configures a default TicketDynmicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that

Description	Value
	will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element3'} = { 'Event' => '', 'Name' => '', 'Value' => '' };`

2.9.5. Ticket::TicketDynamicFieldDefault###Element4

Description	Value
Description:	Configures a default TicketDynmicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element4'} = { 'Event' => '', 'Name' => '', 'Value' => '' };`

2.9.6. Ticket::TicketDynamicFieldDefault###Element5

Description	Value
Description:	Configures a default TicketDynmicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element5'} = { 'Event' => '', 'Name' => '',`

Description	Value
	`  'Value' => ''` `};`

2.9.7. Ticket::TicketDynamicFieldDefault###Element6

Description	Value
Description:	Configures a default TicketDynmicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element6'} =  {`
	`  'Event' => '',`
	`  'Name' => '',`
	`  'Value' => ''`
	`};`

2.9.8. Ticket::TicketDynamicFieldDefault###Element7

Description	Value
Description:	Configures a default TicketDynmicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element7'} =  {`
	`  'Event' => '',`
	`  'Name' => '',`
	`  'Value' => ''`
	`};`

2.9.9. Ticket::TicketDynamicFieldDefault###Element8

Description	Value
Description:	Configures a default TicketDynmicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".

Description	Value
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element8'} = { 'Event' => '', 'Name' => '', 'Value' => '' };`

2.9.10. Ticket::TicketDynamicFieldDefault###Element9

Description	Value
Description:	Configures a default TicketDynmicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element9'} = { 'Event' => '', 'Name' => '', 'Value' => '' };`

2.9.11. Ticket::TicketDynamicFieldDefault###Element10

Description	Value
Description:	Configures a default TicketDynmicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element10'} = { 'Event' => '', 'Name' => '', 'Value' => '' };`

2.9.12. Ticket::TicketDynamicFieldDefault###Element11

Description	Value
Description:	Configures a default TicketDynmicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element11'} = { 'Event' => '', 'Name' => '', 'Value' => '' };`

2.9.13. Ticket::TicketDynamicFieldDefault###Element12

Description	Value
Description:	Configures a default TicketDynmicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element12'} = { 'Event' => '', 'Name' => '', 'Value' => '' };`

2.9.14. Ticket::TicketDynamicFieldDefault###Element13

Description	Value
Description:	Configures a default TicketDynmicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0

Description	Value
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element13'} = { 'Event' => '', 'Name' => '', 'Value' => '' };`

2.9.15. Ticket::TicketDynamicFieldDefault###Element14

Description	Value
Description:	Configures a default TicketDynmicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element14'} = { 'Event' => '', 'Name' => '', 'Value' => '' };`

2.9.16. Ticket::TicketDynamicFieldDefault###Element15

Description	Value
Description:	Configures a default TicketDynmicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element15'} = { 'Event' => '', 'Name' => '', 'Value' => '' };`

2.9.17. Ticket::TicketDynamicFieldDefault###Element16

Description	Value
Description:	Configures a default TicketDynmicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that

Description	Value
	will be set, and "Event" defines the trigger event. Please check the developer manual (http://doc.otrs.org/), chapter "Ticket Event Module".
Group:	Ticket
SubGroup:	Core::TicketDynamicFieldDefault
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element16'} = { 'Event' => '', 'Name' => '', 'Value' => '' };`

2.10. Core::TicketWatcher

2.10.1. Ticket::Watcher

Description	Value
Description:	Enables or disables the ticket watcher feature, to keep track of tickets without being the owner nor the responsible.
Group:	Ticket
SubGroup:	Core::TicketWatcher
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Watcher'} = '0';`

2.10.2. Ticket::WatcherGroup

Description	Value
Description:	Enables ticket watcher feature only for the listed groups.
Group:	Ticket
SubGroup:	Core::TicketWatcher
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::WatcherGroup'} = [ 'admin', 'users' ];`

2.11. Frontend::Admin::ModuleRegistration

2.11.1. Frontend::Module###AdminQueue

Description	Value
Description:	Frontend module registration for the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AdminQueue'} =  {
 'Description' => 'Admin',
 'Group' => [
   'admin'
 ],
 'NavBarModule' => {
   'Block' => 'Queue',
   'Description' => 'Create and manage queues.',
   'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
   'Name' => 'Queues',
   'Prio' => '100'
 },
 'NavBarName' => 'Admin',
 'Title' => 'Queues'
};
```

2.11.2. Frontend::Module###AdminResponse

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AdminResponse'} =  {
 'Description' => 'Admin',
 'Group' => [
   'admin'
 ],
 'NavBarModule' => {
   'Block' => 'Queue',
   'Description' => 'Create and manage response templates.',
   'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
   'Name' => 'Responses',
   'Prio' => '200'
 },
 'NavBarName' => 'Admin',
 'Title' => 'Responses'
};
```

2.11.3. Frontend::Module###AdminQueueResponses

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1

Description	Value
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminQueueResponses'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Queue', 'Description' => 'Link responses to queues.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Responses <-> Queues', 'Prio' => '300' }, 'NavBarName' => 'Admin', 'Title' => 'Responses <-> Queues' };`

2.11.4. Frontend::Module###AdminAutoResponse

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminAutoResponse'} = { 'Description' => 'Admin', 'Group' => ['admin'], 'NavBarModule' => { 'Block' => 'Queue', 'Description' => 'Create and manage responses that are automatically sent.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'Auto Responses', 'Prio' => '400' }, 'NavBarName' => 'Admin', 'Title' => 'Auto Responses' };`

2.11.5. Frontend::Module###AdminQueueAutoResponse

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminQueueAutoResponse'} = {`

Description	Value

	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'NavBarModule' => {`
	`    'Block' => 'Queue',`
	`    'Description' => 'Link queues to auto responses.',`
	`    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`    'Name' => 'Auto Responses <-> Queues',`
	`    'Prio' => '500'`
	`  },`
	`  'NavBarName' => 'Admin',`
	`  'Title' => 'Auto Responses <-> Queues'`
	`};`

2.11.6. Frontend::Module###AdminAttachment

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminAttachment'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'NavBarModule' => {`
	`    'Block' => 'Queue',`
	`    'Description' => 'Create and manage attachments.',`
	`    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`    'Name' => 'Attachments',`
	`    'Prio' => '600'`
	`  },`
	`  'NavBarName' => 'Admin',`
	`  'Title' => 'Attachments'`
	`};`

2.11.7. Frontend::Module###AdminResponseAttachment

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminResponseAttachment'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'NavBarModule' => {`

Description	Value

	`    'Block' => 'Queue',`
	`    'Description' => 'Link attachments to responses templates.',`
	`    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`    'Name' => 'Attachments <-> Responses',`
	`    'Prio' => '700'`
	`  },`
	`  'NavBarName' => 'Admin',`
	`  'Title' => 'Attachments <-> Responses'`
	`};`

2.11.8. Frontend::Module###AdminSalutation

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminSalutation'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'NavBarModule' => {`
	`    'Block' => 'Queue',`
	`    'Description' => 'Create and manage salutations.',`
	`    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`    'Name' => 'Salutations',`
	`    'Prio' => '800'`
	`  },`
	`  'NavBarName' => 'Admin',`
	`  'Title' => 'Salutations'`
	`};`

2.11.9. Frontend::Module###AdminSignature

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminSignature'} =  {`
	`  'Description' => 'Admin',`
	`  'Group' => [`
	`    'admin'`
	`  ],`
	`  'NavBarModule' => {`
	`    'Block' => 'Queue',`
	`    'Description' => 'Create and manage signatures.',`
	`    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`    'Name' => 'Signatures',`
	`    'Prio' => '900'`

Description	Value

	`},`
	`'NavBarName' => 'Admin',`
	`'Title' => 'Signatures'`
	`};`

2.11.10. Frontend::Module###AdminSystemAddress

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminSystemAddress'} = {`
	`'Description' => 'Admin',`
	`'Group' => [`
	`'admin'`
	`],`
	`'NavBarModule' => {`
	`'Block' => 'Email',`
	`'Description' => 'Set sender email addresses for this system.',`
	`'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`'Name' => 'Email Addresses',`
	`'Prio' => '300'`
	`},`
	`'NavBarName' => 'Admin',`
	`'Title' => 'Email Addresses'`
	`};`

2.11.11. Frontend::Module###AdminNotification

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminNotification'} = {`
	`'Description' => 'Admin',`
	`'Group' => [`
	`'admin'`
	`],`
	`'NavBarModule' => {`
	`'Block' => 'Ticket',`
	`'Description' => 'Manage notifications that are sent to agents.',`
	`'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`'Name' => 'Agent Notifications',`
	`'Prio' => '400'`
	`},`
	`'NavBarName' => 'Admin',`
	`'Title' => 'Agent Notifications'`
	`};`

2.11.12. Frontend::Module###AdminNotificationEvent

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AdminNotificationEvent'} =  {
 'Description' => 'Admin',
 'Group' => [
   'admin'
 ],
 'NavBarModule' => {
   'Block' => 'Ticket',
   'Description' => 'Create and manage event based notifications.',
   'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
   'Name' => 'Notifications (Event)',
   'Prio' => '400'
 },
 'NavBarName' => 'Admin',
 'Title' => 'Notifications (Event)'
};
```

2.11.13. Frontend::Module###AdminService

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AdminService'} =  {
 'Description' => 'Admin',
 'Group' => [
   'admin'
 ],
 'NavBarModule' => {
   'Block' => 'Ticket',
   'Description' => 'Create and manage services.',
   'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
   'Name' => 'Services',
   'Prio' => '900'
 },
 'NavBarName' => 'Admin',
 'Title' => 'Services'
};
```

2.11.14. Frontend::Module###AdminSLA

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AdminSLA'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Create and manage Service Level Agreements
 (SLAs).',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Service Level Agreements',
    'Prio' => '1000'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Service Level Agreements'
};
```

2.11.15. Frontend::Module###AdminType

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AdminType'} =  {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Create and manage ticket types.',
    'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',
    'Name' => 'Types',
    'Prio' => '700'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Types'
};
```

2.11.16. Frontend::Module###AdminState

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1

Description	Value
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminState'} = {`
	`'Description' => 'Admin',`
	`'Group' => [`
	`'admin'`
	`],`
	`'NavBarModule' => {`
	`'Block' => 'Ticket',`
	`'Description' => 'Create and manage ticket states.',`
	`'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`'Name' => 'States',`
	`'Prio' => '800'`
	`},`
	`'NavBarName' => 'Admin',`
	`'Title' => 'States'`
	`};`

2.11.17. Frontend::Module###AdminPriority

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminPriority'} = {`
	`'Description' => 'Admin',`
	`'Group' => [`
	`'admin'`
	`],`
	`'NavBarModule' => {`
	`'Block' => 'Ticket',`
	`'Description' => 'Create and manage ticket priorities.',`
	`'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin',`
	`'Name' => 'Priorities',`
	`'Prio' => '850'`
	`},`
	`'NavBarName' => 'Admin',`
	`'Title' => 'Priorities'`
	`};`

2.11.18. Frontend::Module###AdminGenericAgent

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Admin::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AdminGenericAgent'} = {` `'Description' => 'Admin',`

Description	Value
	`'Group' => [ 'admin' ], 'NavBarModule' => { 'Block' => 'System', 'Description' => 'Manage periodic tasks.', 'Module' => 'Kernel::Output::HTML::NavBarModuleAdmin', 'Name' => 'GenericAgent', 'Prio' => '300' }, 'NavBarName' => 'Admin', 'Title' => 'GenericAgent' };`

2.12. Frontend::Agent

2.12.1. Ticket::Frontend::PendingDiffTime

Description	Value
Description:	Time in seconds that gets added to the actual time if setting a pending-state (default: 86400 = 1 day).
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::PendingDiffTime'} = '86400';`

2.12.2. Ticket::Frontend::ListType

Description	Value
Description:	Shows existing parent/child queue lists in the system in the form of a tree or a list.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ListType'} = 'tree';`

2.12.3. Ticket::Frontend::TextAreaEmail

Description	Value
Description:	Permitted width for compose email windows.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'Ticket::Frontend::TextAreaEmail'} = '82';`

2.12.4. Ticket::Frontend::TextAreaNote

Description	Value
Description:	Permitted width for compose note windows.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::TextAreaNote'} = '78';`

2.12.5. Ticket::Frontend::InformAgentMaxSize

Description	Value
Description:	Max size (in rows) of the informed agents box in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::InformAgentMaxSize'} = '3';`

2.12.6. Ticket::Frontend::InvolvedAgentMaxSize

Description	Value
Description:	Max size (in rows) of the involved agents box in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::InvolvedAgentMaxSize'} = '3';`

2.12.7. Ticket::Frontend::CustomerInfoCompose

Description	Value
Description:	Shows the customer user information (phone and email) in the compose screen.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'Ticket::Frontend::CustomerInfoCompose'} = '1';`

2.12.8. Ticket::Frontend::CustomerInfoComposeMaxSize

Description	Value
Description:	Max size (in characters) of the customer information table (phone and email) in the compose screen.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerInfoComposeMaxSize'} = '22';`

2.12.9. Ticket::Frontend::CustomerInfoZoom

Description	Value
Description:	Shows the customer user's info in the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerInfoZoom'} = '1';`

2.12.10. Ticket::Frontend::CustomerInfoZoomMaxSize

Description	Value
Description:	Maximum size (in characters) of the customer information table in the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerInfoZoomMaxSize'} = '22';`

2.12.11. Ticket::Frontend::AccountTime

Description	Value
Description:	Activates time accounting.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1

Description	Value
Config-Setting:	`$Self->{'Ticket::Frontend::AccountTime'} = '1';`

2.12.12. Ticket::Frontend::TimeUnits

Description	Value
Description:	Sets the prefered time units (e.g. work units, hours, minutes).
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::TimeUnits'} = ' (work units)';`

2.12.13. Ticket::Frontend::NeedAccountedTime

Description	Value
Description:	Defines if time accounting is mandatory in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NeedAccountedTime'} = '0';`

2.12.14. Ticket::Frontend::BulkAccountedTime

Description	Value
Description:	Defines if time accounting must be set to all tickets in bulk action.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::BulkAccountedTime'} = '1';`

2.12.15. Ticket::Frontend::NeedSpellCheck

Description	Value
Description:	Defines if composed messages have to be spell checked in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1

Description	Value
Config-Setting:	`$Self->{'Ticket::Frontend::NeedSpellCheck'} = '0';`

2.12.16. Ticket::Frontend::NewOwnerSelection

Description	Value
Description:	Shows an owner selection in phone and email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NewOwnerSelection'} = '1';`

2.12.17. Ticket::Frontend::NewResponsibleSelection

Description	Value
Description:	Show a responsible selection in phone and email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NewResponsibleSelection'} = '1';`

2.12.18. Ticket::Frontend::NewQueueSelectionType

Description	Value
Description:	Defines the receipent target of the phone ticket and the sender of the email ticket ("Queue" shows all queues, "SystemAddress" displays all system addresses) in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NewQueueSelectionType'} = 'Queue';`

2.12.19. Ticket::Frontend::NewQueueSelectionString

Description	Value
Description:	Determines the strings that will be shown as receipent (To:) of the phone ticket and as sender (From:) of the email ticket in the agent interface. For Queue as NewQueueSelectionType "<Queue>" shows

Description	Value
	the names of the queues and for SystemAddress "<Realname> <<Email>>" shows the name and email of the receipent.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::NewQueueSelectionString'} = '<Queue>';`

2.12.20. Ticket::Frontend::NewQueueOwnSelection

Description	Value
Description:	Determines which options will be valid of the recepient (phone ticket) and the sender (email ticket) in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::NewQueueOwnSelection'} = { '1' => 'First Queue!', '2' => 'Second Queue!' };`

2.12.21. Ticket::Frontend::ShowCustomerTickets

Description	Value
Description:	Shows customer history tickets in AgentTicketPhone, AgentTicketEmail and AgentTicketCustomer.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ShowCustomerTickets'} = '1';`

2.12.22. NewTicketInNewWindow::Enabled

Description	Value
Description:	If enabled, TicketPhone and TicketEmail will be open in new windows.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'NewTicketInNewWindow::Enabled'} = '0';`

2.12.23. CustomerDBLink

Description	Value
Description:	Defines an external link to the database of the customer (e.g. 'http://yourhost/customer.php?CID=$Data{"CustomerID"}' or '').
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerDBLink'} = '$Env{"CGIHandle"}?Action=AgentTicketCustomer;TicketID=$Data{"TicketID"}';`

2.12.24. CustomerDBLinkTarget

Description	Value
Description:	Defines the target attribute in the link to external customer database. E.g. 'target="cdb"'.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerDBLinkTarget'} = '';`

2.12.25. Frontend::CommonObject###QueueObject

Description	Value
Description:	Path of the file that stores all the settings for the QueueObject object for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::CommonObject'}->{'QueueObject'} = 'Kernel::System::Queue';`

2.12.26. Frontend::CommonObject###TicketObject

Description	Value
Description:	Path of the file that stores all the settings for the TicketObject for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1

Description	Value
Required:	1
Config-Setting:	`$Self->{'Frontend::CommonObject'}->{'TicketObject'} = 'Kernel::System::Ticket';`

2.12.27. Frontend::CommonParam###Action

Description	Value
Description:	Defines the default used Frontend-Module if no Action parameter given in the url on the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::CommonParam'}->{'Action'} = 'AgentDashboard';`

2.12.28. Frontend::CommonParam###QueueID

Description	Value
Description:	Default queue ID used by the system in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::CommonParam'}->{'QueueID'} = '0';`

2.12.29. Frontend::CommonParam###TicketID

Description	Value
Description:	Default ticket ID used by the system in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Frontend::CommonParam'}->{'TicketID'} = '';`

2.13. Frontend::Agent::CustomerSearch

2.13.1. Ticket::Frontend::CustomerSearchAutoComplete###Active

Description	Value
Description:	Enables or disables the autocomplete feature for the customer search in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::CustomerSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerSearchAutoComplete'}->{'Active'} = '1';`

2.13.2. Ticket::Frontend::CustomerSearchAutoComplete###MinQueryLength

Description	Value
Description:	Sets the minimum number of characters before autocomplete query is sent.
Group:	Ticket
SubGroup:	Frontend::Agent::CustomerSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerSearchAutoComplete'}->{'MinQueryLength'} = '2';`

2.13.3. Ticket::Frontend::CustomerSearchAutoComplete###QueryDelay

Description	Value
Description:	Delay time between autocomplete queries in milliseconds.
Group:	Ticket
SubGroup:	Frontend::Agent::CustomerSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerSearchAutoComplete'}->{'QueryDelay'} = '100';`

2.13.4. Ticket::Frontend::CustomerSearchAutoComplete###MaxResultsDisplayed

Description	Value
Description:	Sets the number of search results to be displayed for the autocomplete feature.
Group:	Ticket
SubGroup:	Frontend::Agent::CustomerSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerSearchAutoComplete'}->{'MaxResultsDisplayed'} = '20';`

2.13.5. Ticket::Frontend::CustomerSearchAutoComplete::DynamicWidth

Description	Value
Description:	Determines if the search results container for the autocomplete feature should adjust its width dynamically.
Group:	Ticket
SubGroup:	Frontend::Agent::CustomerSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerSearchAutoComplete::DynamicWidth'} = '1';`

2.14. Frontend::Agent::Dashboard

2.14.1. DashboardBackend###0100-TicketPendingReminder

Description	Value
Description:	Parameters for the dashboard backend of the ticket pending reminder overview of the agent interface . "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	`$Self->{'DashboardBackend'}->{'0100-TicketPendingReminder'} = { 'Attributes' => 'TicketPendingTimeOlderMinutes=1;StateType=pending reminder;SortBy=PendingTime;OrderBy=Down;', 'Block' => 'ContentLarge', 'CacheTTLLocal' => '0.5', 'Default' => '1', 'Description' => 'All tickets with a reminder set where the reminder date has been reached', 'Filter' => 'Locked', 'Group' => '', 'Limit' => '10', 'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric', 'Permission' => 'rw', 'Time' => 'UntilTime', 'Title' => 'Reminder Tickets' };`

2.14.2. DashboardBackend###0110-TicketEscalation

Description	Value
Description:	Parameters for the dashboard backend of the ticket escalation overview of the agent interface . "Limit" is the number of entries

Description	Value
	shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	`$Self->{'DashboardBackend'}->{'0110-TicketEscalation'} = { 'Attributes' => 'TicketEscalationTimeOlderMinutes=1;SortBy=EscalationTime;OrderBy=Down;', 'Block' => 'ContentLarge', 'CacheTTLLocal' => '0.5', 'Default' => '1', 'Description' => 'All escalated tickets', 'Filter' => 'All', 'Group' => '', 'Limit' => '10', 'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric', 'Permission' => 'rw', 'Time' => 'EscalationTime', 'Title' => 'Escalated Tickets' };`

2.14.3. DashboardBackend###0120-TicketNew

Description	Value
Description:	Parameters for the dashboard backend of the new tickets overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	`$Self->{'DashboardBackend'}->{'0120-TicketNew'} = { 'Attributes' => 'StateType=new;', 'Block' => 'ContentLarge', 'CacheTTLLocal' => '0.5', 'Default' => '1', 'Description' => 'All new tickets, these tickets have not been worked on yet', 'Filter' => 'All', 'Group' => '', 'Limit' => '10', 'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric', 'Permission' => 'rw', 'Time' => 'Age', 'Title' => 'New Tickets' };`

2.14.4. DashboardBackend###0130-TicketOpen

Description	Value
Description:	Parameters for the dashboard backend of the ticket pending reminder overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	`$Self->{'DashboardBackend'}->{'0130-TicketOpen'} = { 'Attributes' => 'StateType=open;', 'Block' => 'ContentLarge', 'CacheTTLLocal' => '0.5', 'Default' => '1', 'Description' => 'All open tickets, these tickets have already been worked on, but need a response', 'Filter' => 'All', 'Group' => '', 'Limit' => '10', 'Module' => 'Kernel::Output::HTML::DashboardTicketGeneric', 'Permission' => 'rw', 'Time' => 'Age', 'Title' => 'Open Tickets / Need to be answered' };`

2.14.5. DashboardBackend###0250-TicketStats

Description	Value
Description:	Parameters for the dashboard backend of the ticket stats of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	`$Self->{'DashboardBackend'}->{'0250-TicketStats'} = { 'Block' => 'ContentSmall', 'CacheTTL' => '30', 'Closed' => '1', 'Created' => '1', 'Default' => '1', 'Group' => '', 'Module' => 'Kernel::Output::HTML::DashboardTicketStatsGeneric', 'Permission' => 'rw', 'Title' => '7 Day Stats'`

Description	Value
	`};`

2.14.6. DashboardBackend###0260-TicketCalendar

Description	Value
Description:	Parameters for the dashboard backend of the ticket calendar of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.
Group:	Ticket
SubGroup:	Frontend::Agent::Dashboard
Valid:	1
Required:	0
Config-Setting:	`$Self->{'DashboardBackend'}->{'0260-TicketCalendar'} = { 'Block' => 'ContentSmall', 'CacheTTL' => '2', 'Default' => '1', 'Group' => '', 'Limit' => '6', 'Module' => 'Kernel::Output::HTML::DashboardCalendar', 'OwnerOnly' => '', 'Permission' => 'rw', 'Title' => 'Upcoming Events' };`

2.15. Frontend::Agent::ModuleMetaHead

2.15.1. Frontend::HeaderMetaModule###2-TicketSearch

Description	Value
Description:	Module to generate html OpenSearch profile for short ticket search in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleMetaHead
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::HeaderMetaModule'}->{'2-TicketSearch'} = { 'Action' => 'AgentTicketSearch', 'Module' => 'Kernel::Output::HTML::HeaderMetaTicketSearch' };`

2.16. Frontend::Agent::ModuleNotify

2.16.1. Frontend::NotifyModule###5-Ticket::TicketEscalation

Description	Value
Description:	Module to show notifications and escalations (ShownMax: max. shown escalations, EscalationInMinutes: Show ticket which will

Description	Value
	escalation in, CacheTime: Cache of calculated escalations in seconds).
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleNotify
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::NotifyModule'}->{'5-Ticket::TicketEscalation'} = { 'CacheTime' => '40', 'EscalationInMinutes' => '120', 'Module' => 'Kernel::Output::HTML::NotificationAgentTicketEscalation', 'ShownMax' => '25' };`

2.17. Frontend::Agent::ModuleRegistration

2.17.1. Frontend::Module###AgentTicketQueue

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketQueue'} = { 'Description' => 'Overview of all open Tickets', 'Loader' => { 'CSS' => ['Core.AgentTicketQueue.css'] }, 'NavBar' => [{ 'AccessKey' => 'o', 'Block' => '', 'Description' => 'Overview of all open Tickets', 'Link' => 'Action=AgentTicketQueue', 'LinkOption' => '', 'Name' => 'Queue view', 'NavBar' => 'Ticket', 'Prio' => '100', 'Type' => '' }, { 'AccessKey' => 't', 'Block' => 'ItemArea', 'Description' => '', 'Link' => 'Action=AgentTicketQueue', 'LinkOption' => '', 'Name' => 'Tickets', 'NavBar' => 'Ticket', 'Prio' => '200', 'Type' => 'Menu' }`

Description	Value

	``
	  ],
	  'NavBarName' => 'Ticket',
	  'Title' => 'QueueView'
	};
	``

2.17.2. Frontend::Module###AgentTicketPhone

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

Config-Setting:	```
	$Self->{'Frontend::Module'}->{'AgentTicketPhone'} = {
	  'Description' => 'Create new phone ticket',
	  'Loader' => {
	    'JavaScript' => [
	      'Core.Agent.CustomerSearch.js',
	      'Core.Agent.TicketAction.js'
	    ]
	  },
	  'NavBar' => [
	    {
	      'AccessKey' => 'n',
	      'Block' => '',
	      'Description' => 'Create new phone ticket (inbound)',
	      'Link' => 'Action=AgentTicketPhone',
	      'LinkOption' => '',
	      'Name' => 'New phone ticket',
	      'NavBar' => 'Ticket',
	      'Prio' => '200',
	      'Type' => ''
	    }
	  ],
	  'NavBarName' => 'Ticket',
	  'Title' => 'New phone ticket'
	};
	``

2.17.3. Frontend::Module###AgentTicketPhoneOutbound

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

Config-Setting:	```
	$Self->{'Frontend::Module'}->{'AgentTicketPhoneOutbound'} = {
	  'Description' => 'Phone Call',
	  'Loader' => {
	    'JavaScript' => [
	      'Core.Agent.TicketAction.js'
	    ]
	  },
	``

Description	Value

	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'Phone-Ticket'`
	`};`

2.17.4. Frontend::Module###AgentTicketPhoneInbound

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AgentTicketPhoneInbound'} =  {
 'Description' => 'Incoming Phone Call',
 'Loader' => {
   'JavaScript' => [
     'Core.Agent.TicketAction.js'
   ]
 },
 'NavBarName' => 'Ticket',
 'Title' => 'Phone-Ticket'
};
```

2.17.5. Frontend::Module###AgentTicketEmail

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Frontend::Module'}->{'AgentTicketEmail'} =  {
 'Description' => 'Create new email ticket',
 'Loader' => {
   'JavaScript' => [
     'Core.Agent.CustomerSearch.js',
     'Core.Agent.TicketAction.js'
   ]
 },
 'NavBar' => [
   {
     'AccessKey' => 'm',
     'Block' => '',
     'Description' => 'Create new email ticket and send this out
(outbound)',
     'Link' => 'Action=AgentTicketEmail',
     'LinkOption' => '',
     'Name' => 'New email ticket',
     'NavBar' => 'Ticket',
     'Prio' => '210',
     'Type' => ''
   }
 ],
 'NavBarName' => 'Ticket',
```

Description	Value
	`  'Title' => 'New email ticket'` `};`

2.17.6. Frontend::Module###AgentTicketSearch

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

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketSearch'} =  {`
	`  'Description' => 'Search Ticket',`
	`  'NavBar' => [`
	`    {`
	`      'AccessKey' => 's',`
	`      'Block' => '',`
	`      'Description' => 'Search Tickets',`
	`      'Link' => 'Action=AgentTicketSearch',`
	`      'LinkOption' => 'onclick="window.setTimeout(function()`
	`{Core.Agent.Search.OpenSearchDialog(\'AgentTicketSearch\');}, 0);`
	` return false;"',`
	`      'Name' => 'Search',`
	`      'NavBar' => 'Ticket',`
	`      'Prio' => '300',`
	`      'Type' => ''`
	`    }`
	`  ],`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'Search'`
	`};`

2.17.7. Frontend::Module###AgentTicketLockedView

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

Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketLockedView'} =  {`
	`  'Description' => 'Locked Tickets',`
	`  'NavBarName' => 'Ticket',`
	`  'Title' => 'Locked Tickets'`
	`};`

2.17.8. Frontend::Module###AgentTicketResponsibleView

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketResponsibleView'} = { 'Description' => 'Responsible Tickets', 'NavBarName' => 'Ticket', 'Title' => 'Responsible Tickets' };`

2.17.9. Frontend::Module###AgentTicketWatchView

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketWatchView'} = { 'Description' => 'Watched Tickets', 'NavBarName' => 'Ticket', 'Title' => 'Watched Tickets' };`

2.17.10. Frontend::Module###AgentCustomerSearch

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentCustomerSearch'} = { 'Description' => 'AgentCustomerSearch', 'NavBarName' => 'Ticket', 'Title' => 'AgentCustomerSearch' };`

2.17.11. Frontend::Module###AgentTicketStatusView

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

Description	Value
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketStatusView'} = { 'Description' => 'Overview of all open tickets', 'NavBar' => [{ 'AccessKey' => 'v', 'Block' => '', 'Description' => 'Overview of all open Tickets.', 'Link' => 'Action=AgentTicketStatusView', 'LinkOption' => '', 'Name' => 'Status view', 'NavBar' => 'Ticket', 'Prio' => '110', 'Type' => '' }], 'NavBarName' => 'Ticket', 'Title' => 'Status view' };`

2.17.12. Frontend::Module###AgentTicketEscalationView

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketEscalationView'} = { 'Description' => 'Overview of all escalated tickets', 'NavBar' => [{ 'AccessKey' => 'e', 'Block' => '', 'Description' => 'Overview Escalated Tickets', 'Link' => 'Action=AgentTicketEscalationView', 'LinkOption' => '', 'Name' => 'Escalation view', 'NavBar' => 'Ticket', 'Prio' => '120', 'Type' => '' }], 'NavBarName' => 'Ticket', 'Title' => 'Escalation view' };`

2.17.13. Frontend::Module###AgentZoom

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

Description	Value
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentZoom'} = { 'Description' => 'compat module for AgentZoom to AgentTicketZoom', 'NavBarName' => 'Ticket', 'Title' => '' };`

2.17.14. Frontend::Module###AgentTicketZoom

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketZoom'} = { 'Description' => 'Ticket Zoom', 'Loader' => { 'JavaScript' => [ 'thirdparty/jquery-tablesorter-2.0.5/jquery.tablesorter.js', 'Core.UI.Table.Sort.js', 'Core.Agent.TicketZoom.js' ] }, 'NavBarName' => 'Ticket', 'Title' => 'Zoom' };`

2.17.15. Frontend::Module###AgentTicketAttachment

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketAttachment'} = { 'Description' => 'To download attachments', 'NavBarName' => 'Ticket', 'Title' => '' };`

2.17.16. Frontend::Module###AgentTicketPlain

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1

Description	Value
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketPlain'} = { 'Description' => 'Ticket plain view of an email', 'NavBarName' => 'Ticket', 'Title' => 'Plain' };`

2.17.17. Frontend::Module###AgentTicketNote

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketNote'} = { 'Description' => 'Ticket Note', 'Loader' => { 'JavaScript' => [ 'Core.Agent.TicketAction.js' ] }, 'NavBarName' => 'Ticket', 'Title' => 'Note' };`

2.17.18. Frontend::Module###AgentTicketMerge

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketMerge'} = { 'Description' => 'Ticket Merge', 'NavBarName' => 'Ticket', 'Title' => 'Merge' };`

2.17.19. Frontend::Module###AgentTicketPending

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1

Description	Value
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketPending'} = { 'Description' => 'Ticket Pending', 'Loader' => { 'JavaScript' => [ 'Core.Agent.TicketAction.js' ] }, 'NavBarName' => 'Ticket', 'Title' => 'Pending' };`

2.17.20. Frontend::Module###AgentTicketWatcher

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketWatcher'} = { 'Description' => 'A TicketWatcher Module', 'NavBarName' => 'Ticket-Watcher', 'Title' => 'Ticket-Watcher' };`

2.17.21. Frontend::Module###AgentTicketPriority

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketPriority'} = { 'Description' => 'Ticket Priority', 'Loader' => { 'JavaScript' => [ 'Core.Agent.TicketAction.js' ] }, 'NavBarName' => 'Ticket', 'Title' => 'Priority' };`

2.17.22. Frontend::Module###AgentTicketLock

Description	Value
Description:	Frontend module registration for the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketLock'} = { 'Description' => 'Ticket Lock', 'NavBarName' => 'Ticket', 'Title' => 'Lock' };`

2.17.23. Frontend::Module###AgentTicketMove

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketMove'} = { 'Description' => 'Ticket Move', 'Loader' => { 'JavaScript' => [ 'Core.Agent.TicketAction.js' ] }, 'NavBarName' => 'Ticket', 'Title' => 'Move' };`

2.17.24. Frontend::Module###AgentTicketHistory

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketHistory'} = { 'Description' => 'Ticket History', 'NavBarName' => 'Ticket', 'Title' => 'History' };`

2.17.25. Frontend::Module###AgentTicketOwner

Description	Value
Description:	Frontend module registration for the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketOwner'} = { 'Description' => 'Ticket Owner', 'Loader' => { 'JavaScript' => [ 'Core.Agent.TicketAction.js' ] }, 'NavBarName' => 'Ticket', 'Title' => 'Owner' };`

2.17.26. Frontend::Module###AgentTicketResponsible

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketResponsible'} = { 'Description' => 'Ticket Responsible', 'Loader' => { 'JavaScript' => [ 'Core.Agent.TicketAction.js' ] }, 'NavBarName' => 'Ticket', 'Title' => 'Responsible' };`

2.17.27. Frontend::Module###AgentTicketCompose

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketCompose'} = { 'Description' => 'Ticket Compose email Answer', 'Loader' => { 'JavaScript' => [ 'Core.Agent.CustomerSearch.js', 'Core.Agent.TicketAction.js'`

Description	Value

	``
	    ]
	  },
	  'NavBarName' => 'Ticket',
	  'Title' => 'Compose'
	};
	``

2.17.28. Frontend::Module###AgentTicketBounce

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

Config-Setting:	```
	$Self->{'Frontend::Module'}->{'AgentTicketBounce'} = {
	  'Description' => 'Ticket Compose Bounce Email',
	  'NavBarName' => 'Ticket',
	  'Title' => 'Bounce'
	};
	``

2.17.29. Frontend::Module###AgentTicketForward

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

Config-Setting:	```
	$Self->{'Frontend::Module'}->{'AgentTicketForward'} = {
	  'Description' => 'Ticket Forward Email',
	  'Loader' => {
	    'JavaScript' => [
	      'Core.Agent.CustomerSearch.js',
	      'Core.Agent.TicketAction.js'
	    ]
	  },
	  'NavBarName' => 'Ticket',
	  'Title' => 'Forward'
	};
	``

2.17.30. Frontend::Module###AgentTicketCustomer

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1

Description	Value
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketCustomer'} = { 'Description' => 'Ticket Customer', 'Loader' => { 'JavaScript' => [ 'Core.Agent.CustomerSearch.js', 'Core.Agent.TicketAction.js' ] }, 'NavBarName' => 'Ticket', 'Title' => 'Customer' };`

2.17.31. Frontend::Module###AgentTicketClose

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketClose'} = { 'Description' => 'Ticket Close', 'Loader' => { 'JavaScript' => [ 'Core.Agent.TicketAction.js' ] }, 'NavBarName' => 'Ticket', 'Title' => 'Close' };`

2.17.32. Frontend::Module###AgentTicketFreeText

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketFreeText'} = { 'Description' => 'Ticket FreeText', 'Loader' => { 'JavaScript' => [ 'Core.Agent.TicketAction.js' ] }, 'NavBarName' => 'Ticket', 'Title' => 'Free Fields' };`

2.17.33. Frontend::Module###AgentTicketPrint

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketPrint'} = { 'Description' => 'Ticket Print', 'NavBarName' => 'Ticket', 'Title' => 'Print' };`

2.17.34. Frontend::Module###AgentTicketBulk

Description	Value
Description:	Frontend module registration for the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Module'}->{'AgentTicketBulk'} = { 'Description' => 'Ticket bulk module', 'Loader' => { 'JavaScript' => [ 'Core.Agent.TicketAction.js' ] }, 'NavBarName' => 'Ticket', 'Title' => 'Bulk-Action' };`

2.18. Frontend::Agent::Preferences

2.18.1. PreferencesGroups###NewTicketNotify

Description	Value
Description:	Parameters for the NewTicketNotify object in the preferences view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	`$Self->{'PreferencesGroups'}->{'NewTicketNotify'} = { 'Active' => '1', 'Column' => 'Email Settings', 'Data' => { '0' => 'No',`

Description	Value

	`    '1' => 'Yes'`
	`  },`
	`  'DataSelected' => '0',`
	`  'Desc' => 'Send me a notification if there is a new ticket in "My Queues".',`
	`  'Key' => 'Send new ticket notifications',`
	`  'Label' => 'New ticket notification',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',`
	`  'PrefKey' => 'UserSendNewTicketNotification',`
	`  'Prio' => '1000'`
	`};`

2.18.2. PreferencesGroups###FollowUpNotify

Description	Value
Description:	Parameters for the FollowUpNotify object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'FollowUpNotify'} =  {`
	`  'Active' => '1',`
	`  'Column' => 'Email Settings',`
	`  'Data' => {`
	`    '0' => 'No',`
	`    '1' => 'Yes'`
	`  },`
	`  'DataSelected' => '0',`
	`  'Desc' => 'Send me a notification if a customer sends a follow up and I\'m the owner of the ticket or the ticket is unlocked and is in one of my subscribed queues.',`
	`  'Key' => 'Send ticket follow up notifications',`
	`  'Label' => 'Ticket follow up notification',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',`
	`  'PrefKey' => 'UserSendFollowUpNotification',`
	`  'Prio' => '2000'`
	`};`

2.18.3. PreferencesGroups###LockTimeoutNotify

Description	Value
Description:	Parameters for the LockTimeoutNotify object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'LockTimeoutNotify'} =  {`
	`  'Active' => '1',`
	`  'Column' => 'Email Settings',`
	`  'Data' => {`

Description	Value

	`    '0' => 'No',`
	`    '1' => 'Yes'`
	`  },`
	`  'DataSelected' => '0',`
	`  'Desc' => 'Send me a notification if a ticket is unlocked by the system.',`
	`  'Key' => 'Send ticket lock timeout notifications',`
	`  'Label' => 'Ticket lock timeout notification',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',`
	`  'PrefKey' => 'UserSendLockTimeoutNotification',`
	`  'Prio' => '3000'`
	`};`

2.18.4. PreferencesGroups###MoveNotify

Description	Value
Description:	Parameters for the MoveNotify object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'MoveNotify'} =  {`
	`  'Active' => '1',`
	`  'Column' => 'Email Settings',`
	`  'Data' => {`
	`    '0' => 'No',`
	`    '1' => 'Yes'`
	`  },`
	`  'DataSelected' => '0',`
	`  'Desc' => 'Send me a notification if a ticket is moved into one of "My Queues".',`
	`  'Key' => 'Send ticket move notifications',`
	`  'Label' => 'Ticket move notification',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',`
	`  'PrefKey' => 'UserSendMoveNotification',`
	`  'Prio' => '4000'`
	`};`

2.18.5. PreferencesGroups###WatcherNotify

Description	Value
Description:	Parameters for the WatcherNotify object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'WatcherNotify'} =  {`
	`  'Active' => '1',`
	`  'Column' => 'Email Settings',`
	`  'Data' => {`

Description	Value

	`    '0' => 'No',`
	`    '1' => 'Yes'`
	`  },`
	`  'DataSelected' => '0',`
	`  'Desc' => 'Send me the same notifications for my watched tickets`
	` that the ticket owners will get.',`
	`  'Key' => 'Send ticket watch notifications',`
	`  'Label' => 'Ticket watch notification',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesTicketWatcher',`
	`  'PrefKey' => 'UserSendWatcherNotification',`
	`  'Prio' => '5000'`
	`};`

2.18.6. PreferencesGroups###CustomQueue

Description	Value
Description:	Parameters for the CustomQueue object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'CustomQueue'} =  {`
	`  'Active' => '1',`
	`  'Column' => 'Other Settings',`
	`  'Desc' => 'Your queue selection of your favorite queues. You also`
	` get notified about those queues via email if enabled.',`
	`  'Key' => 'My Queues',`
	`  'Label' => 'My Queues',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesCustomQueue',`
	`  'Permission' => 'ro',`
	`  'Prio' => '1000'`
	`};`

2.18.7. PreferencesGroups###RefreshTime

Description	Value
Description:	Parameters for the RefreshTime object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'RefreshTime'} =  {`
	`  'Active' => '1',`
	`  'Column' => 'Other Settings',`
	`  'Data' => {`
	`    '0' => 'off',`
	`    '10' => '10 minutes',`
	`    '15' => '15 minutes',`
	`    '2' => ' 2 minutes',`
	`    '5' => ' 5 minutes',`
	`    '7' => ' 7 minutes'`

Description	Value

	``
	  },
	  'DataSelected' => '0',
	  'Desc' => 'If enabled, the different overviews (Dashboard,
	 LockedView, QueueView) will automatically refresh after the specified
	 time.',
	  'Key' => 'Refresh Overviews after',
	  'Label' => 'Overview Refresh Time',
	  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	  'PrefKey' => 'UserRefreshTime',
	  'Prio' => '2000'
	};
	``

2.18.8. PreferencesGroups###TicketOverviewSmallPageShown

Description	Value
Description:	Parameters for the pages (in which the tickets are shown) of the small ticket overview.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	```
	 $Self->{'PreferencesGroups'}->{'TicketOverviewSmallPageShown'} = {
	  'Active' => '0',
	  'Column' => 'Other Settings',
	  'Data' => {
	  '10' => '10',
	  '15' => '15',
	  '20' => '20',
	  '25' => '25',
	  '30' => '30',
	  '35' => '35'
	  },
	  'DataSelected' => '25',
	  'Key' => 'Ticket limit per page for Ticket Overview "Small"',
	  'Label' => 'Ticket Overview "Small" Limit',
	  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	  'PrefKey' => 'UserTicketOverviewSmallPageShown',
	  'Prio' => '8000'
	};
	``

2.18.9. PreferencesGroups###TicketOverviewMediumPageShown

Description	Value
Description:	Parameters for the pages (in which the tickets are shown) of the medium ticket overview.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	```
	 $Self->{'PreferencesGroups'}->{'TicketOverviewMediumPageShown'} = {
	  'Active' => '0',
	  'Column' => 'Other Settings',
	  'Data' => {
	``

Description	Value

	`    '10' => '10',`
	`    '15' => '15',`
	`    '20' => '20',`
	`    '25' => '25',`
	`    '30' => '30',`
	`    '35' => '35'`
	`  },`
	`  'DataSelected' => '20',`
	`  'Key' => 'Ticket limit per page for Ticket Overview "Medium"',`
	`  'Label' => 'Ticket Overview "Medium" Limit',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',`
	`  'PrefKey' => 'UserTicketOverviewMediumPageShown',`
	`  'Prio' => '8100'`
	`};`

2.18.10. PreferencesGroups###TicketOverviewPreviewPageShown

Description	Value
Description:	Parameters for the pages (in which the tickets are shown) of the ticket preview overview.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0

Config-Setting:	`$Self->{'PreferencesGroups'}->{'TicketOverviewPreviewPageShown'} =  {`
	`  'Active' => '0',`
	`  'Column' => 'Other Settings',`
	`  'Data' => {`
	`    '10' => '10',`
	`    '15' => '15',`
	`    '20' => '20',`
	`    '25' => '25',`
	`    '30' => '30',`
	`    '35' => '35'`
	`  },`
	`  'DataSelected' => '15',`
	`  'Key' => 'Ticket limit per page for Ticket Overview "Preview"',`
	`  'Label' => 'Ticket Overview "Preview" Limit',`
	`  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',`
	`  'PrefKey' => 'UserTicketOverviewPreviewPageShown',`
	`  'Prio' => '8200'`
	`};`

2.18.11. PreferencesGroups###CreateNextMask

Description	Value
Description:	Parameters for the CreateNextMask object in the preference view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Preferences
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'PreferencesGroups'}->{'CreateNextMask'} = { 'Active' => '1', 'Column' => 'Other Settings', 'Data' => { '' => 'CreateTicket', 'AgentTicketZoom' => 'TicketZoom' }, 'DataSelected' => '', 'Key' => 'Show this screen after I created a new ticket', 'Label' => 'Screen after new ticket', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserCreateNextMask', 'Prio' => '3000' };`

2.19. Frontend::Agent::SearchRouter

2.19.1. Frontend::Search###Ticket

Description	Value
Description:	Search backend router.
Group:	Ticket
SubGroup:	Frontend::Agent::SearchRouter
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::Search'}->{'Ticket'} = { '^AgentTicket' => 'Action=AgentTicketSearch;Subaction=AJAX' };`

2.20. Frontend::Agent::Ticket::ArticleAttachmentModule

2.20.1. Ticket::Frontend::ArticleAttachmentModule###1-Download

Description	Value
Description:	Shows a link to download article attachments in the zoom view of the article in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleAttachmentModule
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ArticleAttachmentModule'}->{'1-Download'} = { 'Module' => 'Kernel::Output::HTML::ArticleAttachmentDownload' };`

2.20.2. Ticket::Frontend::ArticleAttachmentModule###2-HTML-Viewer

Description	Value
Description:	Shows a link to access article attachments via a html online viewer in the zoom view of the article in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleAttachmentModule
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ArticleAttachmentModule'}->{'2-HTML-Viewer'} = { 'Module' => 'Kernel::Output::HTML::ArticleAttachmentHTMLViewer' };`

2.21. Frontend::Agent::Ticket::ArticleComposeModule

2.21.1. Ticket::Frontend::ArticleComposeModule###1-SignEmail

Description	Value
Description:	Module to compose signed messages (PGP or S/MIME).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleComposeModule
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ArticleComposeModule'}->{'1-SignEmail'} = { 'Module' => 'Kernel::Output::HTML::ArticleComposeSign' };`

2.21.2. Ticket::Frontend::ArticleComposeModule###2-CryptEmail

Description	Value
Description:	Module to crypt composed messages (PGP or S/MIME).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleComposeModule
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ArticleComposeModule'}->{'2-CryptEmail'} = { 'Module' => 'Kernel::Output::HTML::ArticleComposeCrypt' };`

2.22. Frontend::Agent::Ticket::ArticleViewModule

2.22.1. Ticket::Frontend::ArticleViewModule###1-PGP

Description	Value
Description:	Agent interface article notification module to check PGP.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModule

Description	Value
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ArticleViewModule'}->{'1-PGP'} = { 'Module' => 'Kernel::Output::HTML::ArticleCheckPGP' };`

2.22.2. Ticket::Frontend::ArticleViewModule###1-SMIME

Description	Value
Description:	Agent interface module to check incoming emails in the Ticket-Zoom-View if the S/MIME-key is available and true.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModule
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ArticleViewModule'}->{'1-SMIME'} = { 'Module' => 'Kernel::Output::HTML::ArticleCheckSMIME' };`

2.23. Frontend::Agent::Ticket::ArticleViewModulePre

2.23.1. Ticket::Frontend::ArticlePreViewModule###1-PGP

Description	Value
Description:	Agent interface article notification module to check PGP.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModulePre
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ArticlePreViewModule'}->{'1-PGP'} = { 'Module' => 'Kernel::Output::HTML::ArticleCheckPGP' };`

2.23.2. Ticket::Frontend::ArticlePreViewModule###1-SMIME

Description	Value
Description:	Agent interface article notification module to check S/MIME.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ArticleViewModulePre
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ArticlePreViewModule'}->{'1-SMIME'} = { 'Module' => 'Kernel::Output::HTML::ArticleCheckSMIME' };`

2.24. Frontend::Agent::Ticket::MenuModule

2.24.1. Ticket::Frontend::MenuModule###000-Back

Description	Value
Description:	Shows a link in the menu to go back in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'000-Back'} = {`
	`'Action' => '',`
	`'Description' => 'Back',`
	`'Link' => '$Env{"LastScreenOverview"};TicketID=$QData{"TicketID"}',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`'Name' => 'Back',`
	`'PopupType' => '',`
	`'Target' => ''`
	`};`

2.24.2. Ticket::Frontend::MenuModule###100-Lock

Description	Value
Description:	Shows a link in the menu to lock/unlock tickets in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'100-Lock'} = {`
	`'Action' => 'AgentTicketLock',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuLock',`
	`'Name' => 'Lock',`
	`'Target' => ''`
	`};`

2.24.3. Ticket::Frontend::MenuModule###200-History

Description	Value
Description:	Shows a link in the menu to access the history of a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'200-History'} = {`
	`'Action' => 'AgentTicketHistory',`
	`'Description' => 'Show the ticket history',`

Description	Value

	`'Link' => 'Action=AgentTicketHistory;TicketID=$QData{"TicketID"}',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`'Name' => 'History',`
	`'PopupType' => 'TicketHistory',`
	`'Target' => ''`
	`};`

2.24.4. Ticket::Frontend::MenuModule###210-Print

Description	Value
Description:	Shows a link in the menu to print a ticket or an article in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'210-Print'} = {`
	`'Action' => 'AgentTicketPrint',`
	`'Description' => 'Print this ticket',`
	`'Link' => 'Action=AgentTicketPrint;TicketID=$QData{"TicketID"}',`
	`'LinkParam' => 'target="print"',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`'Name' => 'Print',`
	`'PopupType' => 'TicketAction',`
	`'Target' => ''`
	`};`

2.24.5. Ticket::Frontend::MenuModule###300-Priority

Description	Value
Description:	Shows a link in the menu to see the priority of a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'300-Priority'} = {`
	`'Action' => 'AgentTicketPriority',`
	`'Description' => 'Change the ticket priority',`
	`'Link' => 'Action=AgentTicketPriority;TicketID=$QData{"TicketID"}',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`'Name' => 'Priority',`
	`'PopupType' => 'TicketAction',`
	`'Target' => ''`
	`};`

2.24.6. Ticket::Frontend::MenuModule###310-FreeText

Description	Value
Description:	Shows a link in the menu to add a free text field in the ticket zoom view of the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'310-FreeText'} = { 'Action' => 'AgentTicketFreeText', 'Description' => 'Change the free fields for this ticket', 'Link' => 'Action=AgentTicketFreeText;TicketID=$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Free Fields', 'PopupType' => 'TicketAction', 'Target' => '' };`

2.24.7. Ticket::Frontend::MenuModule###320-Link

Description	Value
Description:	Shows a link in the menu that allows linking a ticket with another object in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'320-Link'} = { 'Action' => 'AgentLinkObject', 'Description' => 'Link this ticket to other objects', 'Link' => 'Action=AgentLinkObject;SourceObject=Ticket;SourceKey=$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Link', 'PopupType' => 'TicketAction', 'Target' => '' };`

2.24.8. Ticket::Frontend::MenuModule###400-Owner

Description	Value
Description:	Shows a link in the menu to see the owner of a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'400-Owner'} = { 'Action' => 'AgentTicketOwner', 'Description' => 'Change the owner for this ticket', 'Link' => 'Action=AgentTicketOwner;TicketID=$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`

Description	Value

	`'Name' => 'Owner',`
	`'PopupType' => 'TicketAction',`
	`'Target' => ''`
	`};`

2.24.9. Ticket::Frontend::MenuModule###410-Responsible

Description	Value
Description:	Shows a link in the menu to see the responsible agent of a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'410-Responsible'} = {`
	`'Action' => 'AgentTicketResponsible',`
	`'Description' => 'Change the responsible person for this ticket',`
	`'Link' => 'Action=AgentTicketResponsible;TicketID=`
	`$QData{"TicketID"}',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuResponsible',`
	`'Name' => 'Responsible',`
	`'PopupType' => 'TicketAction',`
	`'Target' => ''`
	`};`

2.24.10. Ticket::Frontend::MenuModule###420-Customer

Description	Value
Description:	Shows a link in the menu to see the customer who requested the ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'420-Customer'} = {`
	`'Action' => 'AgentTicketCustomer',`
	`'Description' => 'Change the customer for this ticket',`
	`'Link' => 'Action=AgentTicketCustomer;TicketID=$QData{"TicketID"}',`
	`'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`'Name' => 'Customer',`
	`'PopupType' => 'TicketAction',`
	`'Target' => ''`
	`};`

2.24.11. Ticket::Frontend::MenuModule###420-Note

Description	Value
Description:	Shows a link in the menu to add a note in the ticket zoom view of the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'420-Note'} = { 'Action' => 'AgentTicketNote', 'Description' => 'Add a note to this ticket', 'Link' => 'Action=AgentTicketNote;TicketID=$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Note', 'PopupType' => 'TicketAction', 'Target' => '' };`

2.24.12. Ticket::Frontend::MenuModule###430-Merge

Description	Value
Description:	Shows a link in the menu that allows merging tickets in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'430-Merge'} = { 'Action' => 'AgentTicketMerge', 'Description' => 'Merge into a different ticket', 'Link' => 'Action=AgentTicketMerge;TicketID=$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Merge', 'PopupType' => 'TicketAction', 'Target' => '' };`

2.24.13. Ticket::Frontend::MenuModule###440-Pending

Description	Value
Description:	Shows a link in the menu to set a ticket as pending in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'440-Pending'} = { 'Action' => 'AgentTicketPending', 'Description' => 'Set this ticket to pending', 'Link' => 'Action=AgentTicketPending;TicketID=$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Pending', 'PopupType' => 'TicketAction', 'Target' => ''`

Description	Value
	`};`

2.24.14. Ticket::Frontend::MenuModule###448-Watch

Description	Value
Description:	Shows a link in the menu for subscribing / unsubscribing from a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Ticket::Frontend::MenuModule'}->{'448-Watch'} =  {
 'Action' => 'AgentTicketWatcher',
 'Module' => 'Kernel::Output::HTML::TicketMenuTicketWatcher',
 'Name' => 'Watch',
 'Target' => ''
};
```

2.24.15. Ticket::Frontend::MenuModule###450-Close

Description	Value
Description:	Shows a link in the menu to close a ticket in the ticket zoom view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'Ticket::Frontend::MenuModule'}->{'450-Close'} =  {
 'Action' => 'AgentTicketClose',
 'Description' => 'Close this ticket',
 'Link' => 'Action=AgentTicketClose;TicketID=$QData{"TicketID"}',
 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',
 'Name' => 'Close',
 'PopupType' => 'TicketAction',
 'Target' => ''
};
```

2.24.16. Ticket::Frontend::MenuModule###460-Delete

Description	Value
Description:	Shows a link in the menu to delete a ticket in the ticket zoom view of the agent interface. Additional access control to show or not show this link can be done by using Key "Group" and Content like "rw:group1;move_into:group2".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	0

Description	Value
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'460-Delete'} = { 'Action' => 'AgentTicketMove', 'Description' => 'Delete this ticket', 'Link' => 'Action=AgentTicketMove;TicketID= $Data{"TicketID"};DestQueue=Delete', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Delete', 'PopupType' => '', 'Target' => '' };`

2.24.17. Ticket::Frontend::MenuModule###470-Spam

Description	Value
Description:	Shows a link to set a ticket as spam in the ticket zoom view of the agent interface. Additional access control to show or not show this link can be done by using Key "Group" and Content like "rw:group1;move_into:group2".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModule
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::MenuModule'}->{'470-Spam'} = { 'Action' => 'AgentTicketMove', 'Description' => 'Mark as Spam!', 'Link' => 'Action=AgentTicketMove;TicketID= $Data{"TicketID"};DestQueue=Delete', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Spam', 'PopupType' => '', 'Target' => '' };`

2.25. Frontend::Agent::Ticket::MenuModulePre

2.25.1. Ticket::Frontend::PreMenuModule###100-Lock

Description	Value
Description:	Shows a link in the menu to lock / unlock a ticket in the ticket overviews of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'100-Lock'} = { 'Action' => 'AgentTicketLock', 'Module' => 'Kernel::Output::HTML::TicketMenuLock', 'Name' => 'Lock', 'PopupType' => '',`

Description	Value
	`  'Target' => ''` `};`

2.25.2. Ticket::Frontend::PreMenuModule###200-Zoom

Description	Value
Description:	Shows a link in the menu to zoom a ticket in the ticket overviews of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'200-Zoom'} =  {`
	`  'Action' => 'AgentTicketZoom',`
	`  'Description' => 'Look into a ticket!',`
	`  'Link' => 'Action=AgentTicketZoom;TicketID=$QData{"TicketID"}',`
	`  'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`  'Name' => 'Zoom',`
	`  'PopupType' => '',`
	`  'Target' => ''`
	`};`

2.25.3. Ticket::Frontend::PreMenuModule###210-History

Description	Value
Description:	Shows a link in the menu to see the history of a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'210-History'} =  {`
	`  'Action' => 'AgentTicketHistory',`
	`  'Description' => 'Show the ticket history',`
	`  'Link' => 'Action=AgentTicketHistory;TicketID=$QData{"TicketID"}',`
	`  'Module' => 'Kernel::Output::HTML::TicketMenuGeneric',`
	`  'Name' => 'History',`
	`  'PopupType' => 'TicketHistory',`
	`  'Target' => ''`
	`};`

2.25.4. Ticket::Frontend::PreMenuModule###300-Priority

Description	Value
Description:	Shows a link in the menu to set the priority of a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre

Description	Value
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'300-Priority'} = { 'Action' => 'AgentTicketPriority', 'Description' => 'Change the priority for this ticket', 'Link' => 'Action=AgentTicketPriority;TicketID=$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Priority', 'PopupType' => 'TicketAction', 'Target' => '' };`

2.25.5. Ticket::Frontend::PreMenuModule###420-Note

Description	Value
Description:	Shows a link in the menu to add a note to a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'420-Note'} = { 'Action' => 'AgentTicketNote', 'Description' => 'Add a note to this ticket', 'Link' => 'Action=AgentTicketNote;TicketID=$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Note', 'PopupType' => 'TicketAction', 'Target' => '' };`

2.25.6. Ticket::Frontend::PreMenuModule###440-Close

Description	Value
Description:	Shows a link in the menu to close a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'440-Close'} = { 'Action' => 'AgentTicketClose', 'Description' => 'Close this ticket', 'Link' => 'Action=AgentTicketClose;TicketID=$QData{"TicketID"}', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Close', 'PopupType' => 'TicketAction', 'Target' => '' };`

2.25.7. Ticket::Frontend::PreMenuModule###445-Move

Description	Value
Description:	Shows a link in the menu to move a ticket in every ticket overview of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'445-Move'} = { 'Action' => 'AgentTicketMove', 'Description' => 'Change queue!', 'Module' => 'Kernel::Output::HTML::TicketMenuMove', 'Name' => 'Move' };`

2.25.8. Ticket::Frontend::PreMenuModule###450-Delete

Description	Value
Description:	Shows a link in the menu to delete a ticket in every ticket overview of the agent interface. Additional access control to show or not show this link can be done by using Key "Group" and Content like "rw:group1;move_into:group2".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'450-Delete'} = { 'Action' => 'AgentTicketMove', 'Description' => 'Delete this ticket', 'Link' => 'Action=AgentTicketMove;TicketID= $Data{"TicketID"};DestQueue=Delete', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Delete', 'PopupType' => '', 'Target' => '' };`

2.25.9. Ticket::Frontend::PreMenuModule###460-Spam

Description	Value
Description:	Shows a link in the menu to set a ticket as spam in every ticket overview of the agent interface. Additional access control to show or not show this link can be done by using Key "Group" and Content like "rw:group1;move_into:group2".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::MenuModulePre
Valid:	0
Required:	0

Description	Value
Config-Setting:	`$Self->{'Ticket::Frontend::PreMenuModule'}->{'460-Spam'} = { 'Action' => 'AgentTicketMove', 'Description' => 'Mark as Spam!', 'Link' => 'Action=AgentTicketMove;TicketID= $Data{"TicketID"};DestQueue=Delete', 'Module' => 'Kernel::Output::HTML::TicketMenuGeneric', 'Name' => 'Spam', 'PopupType' => '', 'Target' => '' };`

2.26. Frontend::Agent::Ticket::ViewBounce

2.26.1. Ticket::Frontend::AgentTicketBounce###Permission

Description	Value
Description:	Required permissions to use the ticket bounce screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'Permission'} = 'bounce';`

2.26.2. Ticket::Frontend::AgentTicketBounce###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket bounce screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'RequiredLock'} = '1';`

2.26.3. Ticket::Frontend::AgentTicketBounce###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after being bounced, in the ticket bounce screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1

Description	Value
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'StateDefault'} = 'closed successful';`

2.26.4. Ticket::Frontend::AgentTicketBounce###StateType

Description	Value
Description:	Defines the next state of a ticket after being bounced, in the ticket bounce screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'StateType'} = [ 'open', 'closed' ];`

2.26.5. Ticket::Frontend::BounceText

Description	Value
Description:	Defines the default ticket bounced notification for customer/sender in the ticket bounce screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBounce
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::BounceText'} = 'Your email with ticket number "<OTRS_TICKET>" is bounced to "<OTRS_BOUNCE_TO>". Contact this address for further information.';`

2.27. Frontend::Agent::Ticket::ViewBulk

2.27.1. Ticket::Frontend::AgentTicketBulk###RequiredLock

Description	Value
Description:	Automatically lock and set owner to current Agent after selecting for an Bulk Action.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'RequiredLock'} = '1';`

2.27.2. Ticket::Frontend::AgentTicketBulk###TicketType

Description	Value
Description:	Sets the ticket type in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'TicketType'} = '1';`

2.27.3. Ticket::Frontend::AgentTicketBulk###Owner

Description	Value
Description:	Sets the ticket owner in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Owner'} = '1';`

2.27.4. Ticket::Frontend::AgentTicketBulk###Responsible

Description	Value
Description:	Sets the responsible agent of the ticket in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Responsible'} = '1';`

2.27.5. Ticket::Frontend::AgentTicketBulk###State

Description	Value
Description:	If a note is added by an agent, sets the state of a ticket in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'State'} = '1';`

2.27.6. Ticket::Frontend::AgentTicketBulk###StateType

Description	Value
Description:	Defines the next state of a ticket after adding a note, in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'StateType'} = [ 'open', 'closed', 'pending reminder', 'pending auto' ];`

2.27.7. Ticket::Frontend::AgentTicketBulk###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after adding a note, in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'StateDefault'} = 'open';`

2.27.8. Ticket::Frontend::AgentTicketBulk###Priority

Description	Value
Description:	Shows the ticket priority options in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Priority'} = '1';`

2.27.9. Ticket::Frontend::AgentTicketBulk###PriorityDefault

Description	Value
Description:	Defines the default ticket priority in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk

Description	Value
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'PriorityDefault'} = '3 normal';`

2.27.10. Ticket::Frontend::AgentTicketBulk###ArticleTypeDefault

Description	Value
Description:	Defines the default type of the note in the ticket bulk screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'ArticleTypeDefault'} = 'note-internal';`

2.27.11. Ticket::Frontend::AgentTicketBulk###ArticleTypes

Description	Value
Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewBulk
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'ArticleTypes'} = { 'note-external' => '1', 'note-internal' => '1', 'note-report' => '0' };`

2.28. Frontend::Agent::Ticket::ViewClose

2.28.1. Ticket::Frontend::AgentTicketClose###Permission

Description	Value
Description:	Required permissions to use the close ticket screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Permission'} = 'close';`

2.28.2. Ticket::Frontend::AgentTicketClose###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the close ticket screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'RequiredLock'} = '1';`

2.28.3. Ticket::Frontend::AgentTicketClose###TicketType

Description	Value
Description:	Sets the ticket type in the close ticket screen of the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'TicketType'} = '0';`

2.28.4. Ticket::Frontend::AgentTicketClose###Service

Description	Value
Description:	Sets the service in the close ticket screen of the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Service'} = '0';`

2.28.5. Ticket::Frontend::AgentTicketClose###Owner

Description	Value
Description:	Sets the ticket owner in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0

Description	Value
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Owner'} =  '0';`

2.28.6. Ticket::Frontend::AgentTicketClose###OwnerMandatory

Description	Value
Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'OwnerMandatory'} = '0';`

2.28.7. Ticket::Frontend::AgentTicketClose###Responsible

Description	Value
Description:	Sets the responsible agent of the ticket in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Responsible'} = '0';`

2.28.8. Ticket::Frontend::AgentTicketClose###State

Description	Value
Description:	If a note is added by an agent, sets the state of a ticket in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'State'} =  '1';`

2.28.9. Ticket::Frontend::AgentTicketClose###StateType

Description	Value
Description:	Defines the next state of a ticket after adding a note, in the close ticket screen of the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'StateType'} = [ 'closed' ];`

2.28.10. Ticket::Frontend::AgentTicketClose###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after adding a note, in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'StateDefault'} = 'closed successful';`

2.28.11. Ticket::Frontend::AgentTicketClose###Note

Description	Value
Description:	Allows adding notes in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Note'} = '1';`

2.28.12. Ticket::Frontend::AgentTicketClose###Subject

Description	Value
Description:	Sets the default subject for notes added in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Subject'} = '$Text{"Close"}';`

2.28.13. Ticket::Frontend::AgentTicketClose###Body

Description	Value
Description:	Sets the default body text for notes added in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Body'} =  '';`

2.28.14. Ticket::Frontend::AgentTicketClose###InvolvedAgent

Description	Value
Description:	Shows a list of all the involved agents on this ticket, in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'InvolvedAgent'} = '0';`

2.28.15. Ticket::Frontend::AgentTicketClose###InformAgent

Description	Value
Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'InformAgent'} = '0';`

2.28.16. Ticket::Frontend::AgentTicketClose###ArticleTypeDefault

Description	Value
Description:	Defines the default type of the note in the close ticket screen of the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ArticleTypeDefault'} = 'note-internal';`

2.28.17. Ticket::Frontend::AgentTicketClose###ArticleTypes

Description	Value
Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };`

2.28.18. Ticket::Frontend::AgentTicketClose###Priority

Description	Value
Description:	Shows the ticket priority options in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Priority'} = '0';`

2.28.19. Ticket::Frontend::AgentTicketClose###PriorityDefault

Description	Value
Description:	Defines the default ticket priority in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'PriorityDefault'} = '3 normal';`

2.28.20. Ticket::Frontend::AgentTicketClose###Title

Description	Value
Description:	Shows the title fields in the close ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Title'} =  '0';`

2.28.21. Ticket::Frontend::AgentTicketClose###HistoryType

Description	Value
Description:	Defines the history type for the close ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'HistoryType'} = 'AddNote';`

2.28.22. Ticket::Frontend::AgentTicketClose###HistoryComment

Description	Value
Description:	Defines the history comment for the close ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'HistoryComment'} = '%%Close';`

2.28.23. Ticket::Frontend::AgentTicketClose###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket close screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewClose
Valid:	1

Description	Value
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketClose'}->{'DynamicField'} = {};`

2.29. Frontend::Agent::Ticket::ViewCompose

2.29.1. Ticket::Frontend::AgentTicketCompose###Permission

Description	Value
Description:	Required permissions to use the ticket compose screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'Permission'} = 'compose';`

2.29.2. Ticket::Frontend::AgentTicketCompose###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket compose screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'RequiredLock'} = '1';`

2.29.3. Ticket::Frontend::AgentTicketCompose###StateDefault

Description	Value
Description:	Defines the default next state of a ticket if it is composed / answered in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'StateDefault'} = 'open';`

2.29.4. Ticket::Frontend::AgentTicketCompose###StateType

Description	Value
Description:	Defines the next possible states after composing / answering a ticket in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'StateType'} = [ 'open', 'closed', 'pending auto', 'pending reminder' ];`

2.29.5. Ticket::Frontend::AgentTicketCompose###ArticleTypes

Description	Value
Description:	Specifies the different article types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'ArticleTypes'} = [ 'email-external', 'email-internal' ];`

2.29.6. Ticket::Frontend::ResponseFormat

Description	Value
Description:	Defines the format of responses in the ticket compose screen of the agent interface ($QData{"OrigFrom"} is From 1:1, $QData{"OrigFromName"} is only realname of From).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1

Config-Setting:	`$Self->{'Ticket::Frontend::ResponseFormat'} = '$QData{"Salutation"} $QData{"StdResponse"} $QData{"Signature"}`
	
	`$TimeShort{"$QData{"Created"}"} - $QData{"OrigFromName"} $Text{"wrote"}: $QData{"Body"}`

Description	Value
	`';`

2.29.7. Ticket::Frontend::Quote

Description	Value
Description:	Defines the used character for email quotes in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::Quote'} =  '>';`

2.29.8. Ticket::Frontend::ComposeAddCustomerAddress

Description	Value
Description:	Adds customers email addresses to recipients in the ticket compose screen of the agent interface. The customers email address won't be added if the article type is email-internal.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ComposeAddCustomerAddress'} =  '1';`

2.29.9. Ticket::Frontend::ComposeReplaceSenderAddress

Description	Value
Description:	Replaces the original sender with current customer's email address on compose answer in the ticket compose screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ComposeReplaceSenderAddress'} =  '0';`

2.29.10. Ticket::Frontend::ComposeExcludeCcRecipients

Description	Value
Description:	Uses Cc recipients in reply Cc list on compose an email answer in the ticket compose screen of the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ComposeExcludeCcRecipients'} =  '0';`

2.29.11. Ticket::Frontend::AgentTicketCompose###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket compose screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCompose
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'DynamicField'} = {};`

2.30. Frontend::Agent::Ticket::ViewCustomer

2.30.1. Ticket::Frontend::AgentTicketCustomer###Permission

Description	Value
Description:	Required permissions to change the customer of a ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCustomer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketCustomer'}->{'Permission'} = 'customer';`

2.30.2. Ticket::Frontend::AgentTicketCustomer###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required to change the customer of a ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewCustomer
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'Ticket::Frontend::AgentTicketCustomer'}->{'RequiredLock'} = '0';`

2.31. Frontend::Agent::Ticket::ViewEmailNew

2.31.1. Ticket::Frontend::AgentTicketEmail###Priority

Description	Value
Description:	Sets the default priority for new email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Priority'} = '3 normal';`

2.31.2. Ticket::Frontend::AgentTicketEmail###ArticleType

Description	Value
Description:	Sets the default article type for new email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'ArticleType'} = 'email-external';`

2.31.3. Ticket::Frontend::AgentTicketEmail###SenderType

Description	Value
Description:	Sets the default sender type for new email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'SenderType'} = 'agent';`

2.31.4. Ticket::Frontend::AgentTicketEmail###Subject

Description	Value
Description:	Sets the default subject for new email tickets (e.g. 'email Outbound') in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Subject'} = '';`

2.31.5. Ticket::Frontend::AgentTicketEmail###Body

Description	Value
Description:	Sets the default text for new email tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Body'} = '';`

2.31.6. Ticket::Frontend::AgentTicketEmail###StateDefault

Description	Value
Description:	Sets the default next ticket state, after the creation of an email ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'StateDefault'} = 'open';`

2.31.7. Ticket::Frontend::AgentTicketEmail###StateType

Description	Value
Description:	Determines the next possible ticket states, after the creation of a new email ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'StateType'} = [ 'open', 'pending auto', 'pending reminder', 'closed' ];`

2.31.8. Ticket::Frontend::AgentTicketEmail###HistoryType

Description	Value
Description:	Defines the history type for the email ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'HistoryType'} = 'EmailAgent';`

2.31.9. Ticket::Frontend::AgentTicketEmail###HistoryComment

Description	Value
Description:	Defines the history comment for the email ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'HistoryComment'} = '';`

2.31.10. Ticket::Frontend::AgentTicketEmail###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket email screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEmailNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'DynamicField'} = {};`

2.32. Frontend::Agent::Ticket::ViewEscalation

2.32.1. Ticket::Frontend::AgentTicketEscalationView###TicketPermission

Description	Value
Description:	Defines the required permission to show a ticket in the escalation view of the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'TicketPermission'} = 'ro';`

2.32.2. Ticket::Frontend::AgentTicketEscalationView###ViewableTicketsPage

Description	Value
Description:	Shows all open tickets (even if they are locked) in the escalation view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'ViewableTicketsPage'} = '50';`

2.32.3. Ticket::Frontend::AgentTicketEscalationView###SortBy::Default

Description	Value
Description:	Defines the default ticket attribute for ticket sorting in the escalation view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'SortBy::Default'} = 'EscalationTime';`

2.32.4. Ticket::Frontend::AgentTicketEscalationView###Order::Default

Description	Value
Description:	Defines the default ticket order (after priority sort) in the escalation view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewEscalation
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'Order::Default'} = 'Up';`

2.33. Frontend::Agent::Ticket::ViewForward

2.33.1. Ticket::Frontend::AgentTicketForward###Permission

Description	Value
Description:	Required permissions to use the ticket forward screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'Permission'} = 'forward';`

2.33.2. Ticket::Frontend::AgentTicketForward###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket forward screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'RequiredLock'} = '1';`

2.33.3. Ticket::Frontend::AgentTicketForward###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after being forwarded, in the ticket forward screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'StateDefault'} = 'closed successful';`

2.33.4. Ticket::Frontend::AgentTicketForward###StateType

Description	Value
Description:	Defines the next possible states after forwarding a ticket in the ticket forward screen of the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'StateType'} = [ 'open', 'closed', 'pending reminder', 'pending auto' ];`

2.33.5. Ticket::Frontend::AgentTicketForward###ArticleTypeDefault

Description	Value
Description:	Defines the default type of forwarded message in the ticket forward screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'ArticleTypeDefault'} = 'email-external';`

2.33.6. Ticket::Frontend::AgentTicketForward###ArticleTypes

Description	Value
Description:	Specifies the different article types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'ArticleTypes'} = [ 'email-external', 'email-internal' ];`

2.33.7. Ticket::Frontend::AgentTicketForward###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket forward screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewForward
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketForward'}->{'DynamicField'} = {};`

2.34. Frontend::Agent::Ticket::ViewFreeText

2.34.1. Ticket::Frontend::AgentTicketFreeText###Permission

Description	Value
Description:	Required permissions to use the ticket free text screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Permission'} = 'rw';`

2.34.2. Ticket::Frontend::AgentTicketFreeText###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket free text screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'RequiredLock'} = '0';`

2.34.3. Ticket::Frontend::AgentTicketFreeText###TicketType

Description	Value
Description:	Sets the ticket type in the ticket free text screen of the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'TicketType'} = '1';`

2.34.4. Ticket::Frontend::AgentTicketFreeText###Service

Description	Value
Description:	Sets the service in the ticket free text screen of the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Service'} = '1';`

2.34.5. Ticket::Frontend::AgentTicketFreeText###Owner

Description	Value
Description:	Sets the ticket owner in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Owner'} = '0';`

2.34.6. Ticket::Frontend::AgentTicketFreeText###OwnerMandatory

Description	Value
Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'OwnerMandatory'} = '0';`

2.34.7. Ticket::Frontend::AgentTicketFreeText###Responsible

Description	Value
Description:	Sets the responsible agent of the ticket in the ticket free text screen of the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Responsible'} = '0';`

2.34.8. Ticket::Frontend::AgentTicketFreeText###State

Description	Value
Description:	If a note is added by an agent, sets the state of a ticket in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'State'} = '0';`

2.34.9. Ticket::Frontend::AgentTicketFreeText###StateType

Description	Value
Description:	Defines the next state of a ticket after adding a note, in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'StateType'} = [ 'open', 'closed', 'pending reminder', 'pending auto' ];`

2.34.10. Ticket::Frontend::AgentTicketFreeText###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after adding a note, in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'StateDefault'} = 'open';`

2.34.11. Ticket::Frontend::AgentTicketFreeText###Note

Description	Value
Description:	Allows adding notes in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Note'} =  '0';`

2.34.12. Ticket::Frontend::AgentTicketFreeText###Subject

Description	Value
Description:	Defines the default subject of a note in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Subject'} = '$Text{"Note"}';`

2.34.13. Ticket::Frontend::AgentTicketFreeText###Body

Description	Value
Description:	Defines the default body of a note in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Body'} =  '';`

2.34.14. Ticket::Frontend::AgentTicketFreeText###InvolvedAgent

Description	Value
Description:	Shows a list of all the involved agents on this ticket, in the ticket free text screen of the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'InvolvedAgent'} = '0';`

2.34.15. Ticket::Frontend::AgentTicketFreeText###InformAgent

Description	Value
Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'InformAgent'} = '0';`

2.34.16. Ticket::Frontend::AgentTicketFreeText###ArticleTypeDefault

Description	Value
Description:	Defines the default type of the note in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ArticleTypeDefault'} = 'note-internal';`

2.34.17. Ticket::Frontend::AgentTicketFreeText###ArticleTypes

Description	Value
Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ArticleTypes'} = { 'note-external' => '1',`

Description	Value

	`'note-internal' => '1',`
	`'note-report' => '0'`
	`};`

2.34.18. Ticket::Frontend::AgentTicketFreeText###Priority

Description	Value
Description:	Shows the ticket priority options in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Priority'} = '0';`

2.34.19. Ticket::Frontend::AgentTicketFreeText###PriorityDefault

Description	Value
Description:	Defines the default ticket priority in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'PriorityDefault'} = '3 normal';`

2.34.20. Ticket::Frontend::AgentTicketFreeText###Title

Description	Value
Description:	Shows the title fields in the ticket free text screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Title'} = '1';`

2.34.21. Ticket::Frontend::AgentTicketFreeText###HistoryType

Description	Value
Description:	Defines the history type for the ticket free text screen action, which gets used for ticket history.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'HistoryType'} = 'AddNote';`

2.34.22. Ticket::Frontend::AgentTicketFreeText###HistoryComment

Description	Value
Description:	Defines the history comment for the ticket free text screen action, which gets used for ticket history.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'HistoryComment'} = '%%FreeText';`

2.34.23. Ticket::Frontend::AgentTicketFreeText###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket free text screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewFreeText
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'DynamicField'} = {};`

2.35. Frontend::Agent::Ticket::ViewHistory

2.35.1. Ticket::Frontend::HistoryOrder

Description	Value
Description:	Shows the ticket history (reverse ordered) in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewHistory
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'Ticket::Frontend::HistoryOrder'} = 'normal';`

2.36. Frontend::Agent::Ticket::ViewMailbox

2.36.1. Ticket::Frontend::AgentTicketLockedView###SortBy::Default

Description	Value
Description:	Defines the default ticket attribute for ticket sorting in the locked ticket view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketLockedView'}->{'SortBy::Default'} = 'Age';`

2.36.2. Ticket::Frontend::AgentTicketLockedView###Order::Default

Description	Value
Description:	Defines the default ticket order in the ticket locked view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketLockedView'}->{'Order::Default'} = 'Up';`

2.36.3. Ticket::Frontend::AgentTicketResponsibleView###SortBy::Default

Description	Value
Description:	Defines the default ticket attribute for ticket sorting in the responsible view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsibleView'}->{'SortBy::Default'} = 'Age';`

2.36.4. Ticket::Frontend::AgentTicketResponsibleView###Order::Default

Description	Value
Description:	Defines the default ticket order in the responsible view of the agent interface. Up: oldest on top. Down: latest on top.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsibleView'}->{'Order::Default'} = 'Up';`

2.36.5. Ticket::Frontend::AgentTicketWatchView###SortBy::Default

Description	Value
Description:	Defines the default ticket attribute for ticket sorting in the watch view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketWatchView'}->{'SortBy::Default'} = 'Age';`

2.36.6. Ticket::Frontend::AgentTicketWatchView###Order::Default

Description	Value
Description:	Defines the default ticket order in the watch view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMailbox
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketWatchView'}->{'Order::Default'} = 'Up';`

2.37. Frontend::Agent::Ticket::ViewMerge

2.37.1. Ticket::Frontend::AgentTicketMerge###Permission

Description	Value
Description:	Required permissions to use the ticket merge screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	1

Description	Value
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'Permission'} = 'rw';`

2.37.2. Ticket::Frontend::AgentTicketMerge###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket merge screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'RequiredLock'} = '1';`

2.37.3. Ticket::Frontend::MergeText

Description	Value
Description:	When tickets are merged, the customer can be informed per email by setting the check box "Inform Sender". In this text area, you can define a pre-formatted text which can later be modified by the agents.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::MergeText'} =  'Your email with ticket number "<OTRS_TICKET>" is merged to "<OTRS_MERGE_TO_TICKET>".';`

2.37.4. Ticket::Frontend::AutomaticMergeText

Description	Value
Description:	When tickets are merged, a note will be added automatically to the ticket which is no longer active. In this text area you can define this text (This text cannot be changed by the agent).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMerge
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AutomaticMergeText'} =  'Merged Ticket <OTRS_TICKET> to <OTRS_MERGE_TO_TICKET>.';`

2.38. Frontend::Agent::Ticket::ViewMove

2.38.1. Ticket::Frontend::MoveType

Description	Value
Description:	Determines if the list of possible queues to move to ticket into should be displayed in a dropdown list or in a new window in the agent interface. If "New Window" is set you can add a move note to the ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::MoveType'} = 'form';`

2.38.2. Ticket::Frontend::AgentTicketMove###State

Description	Value
Description:	Allows to set a new ticket state in the move ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'State'} = '1';`

2.38.3. Ticket::DefaultNextMoveStateType

Description	Value
Description:	Defines the next state of a ticket after being moved to another queue, in the move ticket screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::DefaultNextMoveStateType'} = [ 'open', 'closed' ];`

2.38.4. Ticket::Frontend::AgentTicketMove###Priority

Description	Value
Description:	Shows the ticket priority options in the move ticket screen of the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Priority'} =  '0';`

2.38.5. Ticket::Frontend::AgentTicketMove###NextScreen

Description	Value
Description:	Determines the next screen after the ticket is moved. LastScreenOverview will return to search results, queueview, dashboard or the like, LastScreenView will return to TicketZoom.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'NextScreen'} = 'LastScreenView';`

2.38.6. Ticket::Frontend::AgentTicketMove###Subject

Description	Value
Description:	Sets the default subject for notes added in the ticket move screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Subject'} = '$Text{"Change Queue"}';`

2.38.7. Ticket::Frontend::AgentTicketMove###Body

Description	Value
Description:	Sets the default body text for notes added in the ticket move screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Body'} =  '';`

2.38.8. Ticket::Frontend::AgentTicketMove###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket move screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewMove
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketMove'}->{'DynamicField'} = {};`

2.39. Frontend::Agent::Ticket::ViewNote

2.39.1. Ticket::Frontend::AgentTicketNote###Permission

Description	Value
Description:	Required permissions to use the ticket note screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Permission'} = 'note';`

2.39.2. Ticket::Frontend::AgentTicketNote###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket note screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'RequiredLock'} = '0';`

2.39.3. Ticket::Frontend::AgentTicketNote###TicketType

Description	Value
Description:	Sets the ticket type in the ticket note screen of the agent interface (Ticket::Type needs to be activated).
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'TicketType'} = '0';`

2.39.4. Ticket::Frontend::AgentTicketNote###Service

Description	Value
Description:	Sets the service in the ticket note screen of the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Service'} = '0';`

2.39.5. Ticket::Frontend::AgentTicketNote###Owner

Description	Value
Description:	Sets the ticket owner in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Owner'} = '0';`

2.39.6. Ticket::Frontend::AgentTicketNote###OwnerMandatory

Description	Value
Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'OwnerMandatory'} = '0';`

2.39.7. Ticket::Frontend::AgentTicketNote###Responsible

Description	Value
Description:	Sets the responsible agent of the ticket in the ticket note screen of the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Responsible'} =  '0';`

2.39.8. Ticket::Frontend::AgentTicketNote###State

Description	Value
Description:	If a note is added by an agent, sets the state of a ticket in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'State'} =  '0';`

2.39.9. Ticket::Frontend::AgentTicketNote###StateType

Description	Value
Description:	Defines the next state of a ticket after adding a note, in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'StateType'} =  [ 'open', 'closed', 'pending reminder', 'pending auto' ];`

2.39.10. Ticket::Frontend::AgentTicketNote###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after adding a note, in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'StateDefault'} = 'open';`

2.39.11. Ticket::Frontend::AgentTicketNote###Note

Description	Value
Description:	Allows adding notes in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Note'} =  '1';`

2.39.12. Ticket::Frontend::AgentTicketNote###Subject

Description	Value
Description:	Sets the default subject for notes added in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Subject'} = '$Text{"Note"}';`

2.39.13. Ticket::Frontend::AgentTicketNote###Body

Description	Value
Description:	Sets the default body text for notes added in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Body'} =  '';`

2.39.14. Ticket::Frontend::AgentTicketNote###InvolvedAgent

Description	Value
Description:	Shows a list of all the involved agents on this ticket, in the ticket note screen of the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'InvolvedAgent'} = '0';`

2.39.15. Ticket::Frontend::AgentTicketNote###InformAgent

Description	Value
Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'InformAgent'} = '0';`

2.39.16. Ticket::Frontend::AgentTicketNote###ArticleTypeDefault

Description	Value
Description:	Defines the default type of the note in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'ArticleTypeDefault'} = 'note-internal';`

2.39.17. Ticket::Frontend::AgentTicketNote###ArticleTypes

Description	Value
Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'ArticleTypes'} = { 'note-external' => '1', 'note-internal' => '1',`

Description	Value
	`  'note-report' => '0'` `};`

2.39.18. Ticket::Frontend::AgentTicketNote###Priority

Description	Value
Description:	Shows the ticket priority options in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Priority'} =  '0';`

2.39.19. Ticket::Frontend::AgentTicketNote###PriorityDefault

Description	Value
Description:	Defines the default ticket priority in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'PriorityDefault'} = '3 normal';`

2.39.20. Ticket::Frontend::AgentTicketNote###Title

Description	Value
Description:	Shows the title fields in the ticket note screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Title'} =  '0';`

2.39.21. Ticket::Frontend::AgentTicketNote###HistoryType

Description	Value
Description:	Defines the history type for the ticket note screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote

Description	Value
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'HistoryType'} = 'AddNote';`

2.39.22. Ticket::Frontend::AgentTicketNote###HistoryComment

Description	Value
Description:	Defines the history comment for the ticket note screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'HistoryComment'} = '%%Note';`

2.39.23. Ticket::Frontend::AgentTicketNote###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket note screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewNote
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketNote'}->{'DynamicField'} = {};`

2.40. Frontend::Agent::Ticket::ViewOwner

2.40.1. Ticket::Frontend::AgentTicketOwner###Permission

Description	Value
Description:	Required permissions to use the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Permission'} = 'owner';`

2.40.2. Ticket::Frontend::AgentTicketOwner###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket owner screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'RequiredLock'} = '0';`

2.40.3. Ticket::Frontend::AgentTicketOwner###TicketType

Description	Value
Description:	Sets the ticket type in the ticket owner screen of a zoomed ticket in the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'TicketType'} = '0';`

2.40.4. Ticket::Frontend::AgentTicketOwner###Service

Description	Value
Description:	Sets the service in the ticket owner screen of a zoomed ticket in the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Service'} = '0';`

2.40.5. Ticket::Frontend::AgentTicketOwner###Owner

Description	Value
Description:	Sets the ticket owner in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1

Description	Value
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Owner'} =  '1';`

2.40.6. Ticket::Frontend::AgentTicketOwner###OwnerMandatory

Description	Value
Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'OwnerMandatory'} = '1';`

2.40.7. Ticket::Frontend::AgentTicketOwner###Responsible

Description	Value
Description:	Sets the responsible agent of the ticket in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Responsible'} = '0';`

2.40.8. Ticket::Frontend::AgentTicketOwner###State

Description	Value
Description:	If a note is added by an agent, sets the state of the ticket in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'State'} =  '0';`

2.40.9. Ticket::Frontend::AgentTicketOwner###StateType

Description	Value
Description:	Defines the next state of a ticket after adding a note, in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'StateType'} = [ 'open', 'pending reminder', 'pending auto' ];`

2.40.10. Ticket::Frontend::AgentTicketOwner###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after adding a note, in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'StateDefault'} = 'open';`

2.40.11. Ticket::Frontend::AgentTicketOwner###Note

Description	Value
Description:	Allows adding notes in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Note'} = '1';`

2.40.12. Ticket::Frontend::AgentTicketOwner###Subject

Description	Value
Description:	Sets the default subject for notes added in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Subject'} = '$Text{"Owner Update"}!';`

2.40.13. Ticket::Frontend::AgentTicketOwner###Body

Description	Value
Description:	Sets the default body text for notes added in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Body'} =  '';`

2.40.14. Ticket::Frontend::AgentTicketOwner###InvolvedAgent

Description	Value
Description:	Shows a list of all the involved agents on this ticket, in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'InvolvedAgent'} = '0';`

2.40.15. Ticket::Frontend::AgentTicketOwner###InformAgent

Description	Value
Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'InformAgent'} = '0';`

2.40.16. Ticket::Frontend::AgentTicketOwner###ArticleTypeDefault

Description	Value
Description:	Defines the default type of the note in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'ArticleTypeDefault'} = 'note-internal';`

2.40.17. Ticket::Frontend::AgentTicketOwner###ArticleTypes

Description	Value
Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };`

2.40.18. Ticket::Frontend::AgentTicketOwner###Priority

Description	Value
Description:	Shows the ticket priority options in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Priority'} = '0';`

2.40.19. Ticket::Frontend::AgentTicketOwner###PriorityDefault

Description	Value
Description:	Defines the default ticket priority in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'PriorityDefault'} = '3 normal';`

2.40.20. Ticket::Frontend::AgentTicketOwner###Title

Description	Value
Description:	Shows the title fields in the ticket owner screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Title'} =  '0';`

2.40.21. Ticket::Frontend::AgentTicketOwner###HistoryType

Description	Value
Description:	Defines the history type for the ticket owner screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'HistoryType'} = 'AddNote';`

2.40.22. Ticket::Frontend::AgentTicketOwner###HistoryComment

Description	Value
Description:	Defines the history comment for the ticket owner screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'HistoryComment'} = '%%Owner';`

2.40.23. Ticket::Frontend::AgentTicketOwner###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket owner screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewOwner

Description	Value
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'DynamicField'} = {};`

2.41. Frontend::Agent::Ticket::ViewPending

2.41.1. Ticket::Frontend::AgentTicketPending###Permission

Description	Value
Description:	Required permissions to use the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Permission'} = 'pending';`

2.41.2. Ticket::Frontend::AgentTicketPending###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket pending screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'RequiredLock'} = '1';`

2.41.3. Ticket::Frontend::AgentTicketPending###TicketType

Description	Value
Description:	Sets the ticket type in the ticket pending screen of a zoomed ticket in the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'TicketType'} = '0';`

2.41.4. Ticket::Frontend::AgentTicketPending###Service

Description	Value
Description:	Sets the service in the ticket pending screen of a zoomed ticket in the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Service'} =  '0';`

2.41.5. Ticket::Frontend::AgentTicketPending###Owner

Description	Value
Description:	Sets the ticket owner in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Owner'} =  '0';`

2.41.6. Ticket::Frontend::AgentTicketPending###OwnerMandatory

Description	Value
Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'OwnerMandatory'} = '0';`

2.41.7. Ticket::Frontend::AgentTicketPending###Responsible

Description	Value
Description:	Sets the responsible agent of the ticket in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Responsible'} = '0';`

2.41.8. Ticket::Frontend::AgentTicketPending###State

Description	Value
Description:	If a note is added by an agent, sets the state of the ticket in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'State'} =  '1';`

2.41.9. Ticket::Frontend::AgentTicketPending###StateType

Description	Value
Description:	Defines the next state of a ticket after adding a note, in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'StateType'} =  [ 'pending reminder', 'pending auto' ];`

2.41.10. Ticket::Frontend::AgentTicketPending###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after adding a note, in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'StateDefault'} = 'pending reminder';`

2.41.11. Ticket::Frontend::AgentTicketPending###Note

Description	Value
Description:	Allows adding notes in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Note'} =  '1';`

2.41.12. Ticket::Frontend::AgentTicketPending###Subject

Description	Value
Description:	Sets the default subject for notes added in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Subject'} = '$Text{"Pending"}!';`

2.41.13. Ticket::Frontend::AgentTicketPending###Body

Description	Value
Description:	Sets the default body text for notes added in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Body'} =  '';`

2.41.14. Ticket::Frontend::AgentTicketPending###InvolvedAgent

Description	Value
Description:	Shows a list of all the involved agents on this ticket, in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0

Description	Value
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'InvolvedAgent'} = '0';`

2.41.15. Ticket::Frontend::AgentTicketPending###InformAgent

Description	Value
Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'InformAgent'} = '0';`

2.41.16. Ticket::Frontend::AgentTicketPending###ArticleTypeDefault

Description	Value
Description:	Defines the default type of the note in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'ArticleTypeDefault'} = 'note-internal';`

2.41.17. Ticket::Frontend::AgentTicketPending###ArticleTypes

Description	Value
Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0'`

Description	Value
	`};`

2.41.18. Ticket::Frontend::AgentTicketPending###Priority

Description	Value
Description:	Shows the ticket priority options in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Priority'} = '0';`

2.41.19. Ticket::Frontend::AgentTicketPending###PriorityDefault

Description	Value
Description:	Defines the default ticket priority in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'PriorityDefault'} = '3 normal';`

2.41.20. Ticket::Frontend::AgentTicketPending###Title

Description	Value
Description:	Shows the title fields in the ticket pending screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Title'} = '0';`

2.41.21. Ticket::Frontend::AgentTicketPending###HistoryType

Description	Value
Description:	Defines the history type for the ticket pending screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending

Description	Value
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'HistoryType'} = 'AddNote';`

2.41.22. Ticket::Frontend::AgentTicketPending###HistoryComment

Description	Value
Description:	Defines the history comment for the ticket pending screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'HistoryComment'} = '%%Pending';`

2.41.23. Ticket::Frontend::AgentTicketPending###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket pending screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPending
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPending'}->{'DynamicField'} = {};`

2.42. Frontend::Agent::Ticket::ViewPhoneInbound

2.42.1. Ticket::Frontend::AgentTicketPhoneInbound###Permission

Description	Value
Description:	Required permissions to use the ticket phone inbound screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'Permission'} = 'phone';`

2.42.2. Ticket::Frontend::AgentTicketPhoneInbound###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket phone inbound screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'RequiredLock'} = '0';`

2.42.3. Ticket::Frontend::AgentTicketPhoneInbound###ArticleType

Description	Value
Description:	Defines the default type of the note in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'ArticleType'} = 'phone';`

2.42.4. Ticket::Frontend::AgentTicketPhoneInbound###SenderType

Description	Value
Description:	Defines the default sender type for phone tickets in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'SenderType'} = 'customer';`

2.42.5. Ticket::Frontend::AgentTicketPhoneInbound###Subject

Description	Value
Description:	Defines the default subject for phone tickets in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound

Description	Value
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'Subject'} = '$Text{"Phone call"}!';`

2.42.6. Ticket::Frontend::AgentTicketPhoneInbound###Body

Description	Value
Description:	Defines the default note body text for phone tickets in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'Body'} =  '';`

2.42.7. Ticket::Frontend::AgentTicketPhoneInbound###State

Description	Value
Description:	Defines the default ticket next state after adding a phone note in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'State'} = 'open';`

2.42.8. Ticket::Frontend::AgentTicketPhoneInbound###StateType

Description	Value
Description:	Next possible ticket states after adding a phone note in the ticket phone inbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'StateType'} = [ 'open', 'pending auto', 'pending reminder', 'closed'`

Description	Value
	`];`

2.42.9. Ticket::Frontend::AgentTicketPhoneInbound###HistoryType

Description	Value
Description:	Defines the history type for the ticket phone inbound screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'HistoryType'} = 'PhoneCallCustomer';`

2.42.10. Ticket::Frontend::AgentTicketPhoneInbound###HistoryComment

Description	Value
Description:	Defines the history comment for the ticket phone inbound screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'HistoryComment'} = '';`

2.42.11. Ticket::Frontend::AgentTicketPhoneInbound###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket phone inbound screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneInbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'DynamicField'} = {};`

2.43. Frontend::Agent::Ticket::ViewPhoneNew

2.43.1. Ticket::Frontend::AgentTicketPhone###Priority

Description	Value
Description:	Sets the default priority for new phone tickets in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'Priority'} = '3 normal';`

2.43.2. Ticket::Frontend::AgentTicketPhone###ArticleType

Description	Value
Description:	Sets the default article type for new phone tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'ArticleType'} = 'phone';`

2.43.3. Ticket::Frontend::AgentTicketPhone###SenderType

Description	Value
Description:	Sets the default sender type for new phone ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'SenderType'} = 'customer';`

2.43.4. Ticket::Frontend::AgentTicketPhone::AllowMultipleFrom

Description	Value
Description:	Controls if more than one from entry can be set in the new phone ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'Ticket::Frontend::AgentTicketPhone::AllowMultipleFrom'} = '1';`

2.43.5. Ticket::Frontend::AgentTicketPhone###Subject

Description	Value
Description:	Sets the default subject for new phone tickets (e.g. 'Phone call') in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'Subject'} = '';`

2.43.6. Ticket::Frontend::AgentTicketPhone###Body

Description	Value
Description:	Sets the default note text for new telephone tickets. E.g 'New ticket via call' in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'Body'} = '';`

2.43.7. Ticket::Frontend::AgentTicketPhone###StateDefault

Description	Value
Description:	Sets the default next state for new phone tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'StateDefault'} = 'open';`

2.43.8. Ticket::Frontend::AgentTicketPhone###StateType

Description	Value
Description:	Determines the next possible ticket states, after the creation of a new phone ticket in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'StateType'} = [ 'open', 'pending auto', 'pending reminder', 'closed' ];`

2.43.9. Ticket::Frontend::AgentTicketPhone###HistoryType

Description	Value
Description:	Defines the history type for the phone ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'HistoryType'} = 'PhoneCallCustomer';`

2.43.10. Ticket::Frontend::AgentTicketPhone###HistoryComment

Description	Value
Description:	Defines the history comment for the phone ticket screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'HistoryComment'} = '';`

2.43.11. Ticket::Frontend::AgentTicketPhone###SplitLinkType

Description	Value
Description:	Sets the default link type of splitted tickets in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0

Description	Value
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'SplitLinkType'} = { 'Direction' => 'Target', 'LinkType' => 'ParentChild' };`

2.43.12. Ticket::Frontend::AgentTicketPhone###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket phone screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'DynamicField'} = {};`

2.44. Frontend::Agent::Ticket::ViewPhoneOutbound

2.44.1. Ticket::Frontend::AgentTicketPhoneOutbound###Permission

Description	Value
Description:	Required permissions to use the ticket phone outbound screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'Permission'} = 'phone';`

2.44.2. Ticket::Frontend::AgentTicketPhoneOutbound###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket phone outbound screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'RequiredLock'} = '1';`

2.44.3. Ticket::Frontend::AgentTicketPhoneOutbound###ArticleType

Description	Value
Description:	Defines the default type of the note in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'ArticleType'} = 'phone';`

2.44.4. Ticket::Frontend::AgentTicketPhoneOutbound###SenderType

Description	Value
Description:	Defines the default sender type for phone tickets in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'SenderType'} = 'agent';`

2.44.5. Ticket::Frontend::AgentTicketPhoneOutbound###Subject

Description	Value
Description:	Defines the default subject for phone tickets in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'Subject'} = '$Text{"Phone call"}!';`

2.44.6. Ticket::Frontend::AgentTicketPhoneOutbound###Body

Description	Value
Description:	Defines the default note body text for phone tickets in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1

Description	Value
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'Body'} = '';`

2.44.7. Ticket::Frontend::AgentTicketPhoneOutbound###State

Description	Value
Description:	Defines the default ticket next state after adding a phone note in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'State'} = 'closed successful';`

2.44.8. Ticket::Frontend::AgentTicketPhoneOutbound###StateType

Description	Value
Description:	Next possible ticket states after adding a phone note in the ticket phone outbound screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'StateType'} = [ 'open', 'pending auto', 'pending reminder', 'closed' ];`

2.44.9. Ticket::Frontend::AgentTicketPhoneOutbound###HistoryType

Description	Value
Description:	Defines the history type for the ticket phone outbound screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'HistoryType'} = 'PhoneCallAgent';`

2.44.10. Ticket::Frontend::AgentTicketPhoneOutbound###HistoryComment

Description	Value
Description:	Defines the history comment for the ticket phone outbound screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'HistoryComment'} = '';`

2.44.11. Ticket::Frontend::AgentTicketPhoneOutbound###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket phone outbound screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPhoneOutbound
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'DynamicField'} = {};`

2.45. Frontend::Agent::Ticket::ViewPrint

2.45.1. Ticket::Frontend::AgentTicketPrint###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket print screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPrint
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPrint'}->{'DynamicField'} = {};`

2.46. Frontend::Agent::Ticket::ViewPriority

2.46.1. Ticket::Frontend::AgentTicketPriority###Permission

Description	Value
Description:	Required permissions to use the ticket priority screen of a zoomed ticket in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Permission'} = 'priority';`

2.46.2. Ticket::Frontend::AgentTicketPriority###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket priority screen of a zoomed ticket in the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'RequiredLock'} = '1';`

2.46.3. Ticket::Frontend::AgentTicketPriority###TicketType

Description	Value
Description:	Sets the ticket type in the ticket priority screen of a zoomed ticket in the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'TicketType'} = '0';`

2.46.4. Ticket::Frontend::AgentTicketPriority###Service

Description	Value
Description:	Sets the service in the ticket priority screen of a zoomed ticket in the agent interface (Ticket::Service needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0

Description	Value
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Service'} = '0';`

2.46.5. Ticket::Frontend::AgentTicketPriority###Owner

Description	Value
Description:	Sets the ticket owner in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Owner'} = '0';`

2.46.6. Ticket::Frontend::AgentTicketPriority###OwnerMandatory

Description	Value
Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'OwnerMandatory'} = '0';`

2.46.7. Ticket::Frontend::AgentTicketPriority###Responsible

Description	Value
Description:	Sets the responsible agent of the ticket in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Responsible'} = '0';`

2.46.8. Ticket::Frontend::AgentTicketPriority###State

Description	Value
Description:	If a note is added by an agent, sets the state of the ticket in the ticket priority screen of a zoomed ticket in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'State'} = '0';`

2.46.9. Ticket::Frontend::AgentTicketPriority###StateType

Description	Value
Description:	Defines the next state of a ticket after adding a note, in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'StateType'} = [ 'open', 'pending reminder', 'pending auto' ];`

2.46.10. Ticket::Frontend::AgentTicketPriority###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after adding a note, in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'StateDefault'} = 'open';`

2.46.11. Ticket::Frontend::AgentTicketPriority###Note

Description	Value
Description:	Allows adding notes in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Note'} =  '1';`

2.46.12. Ticket::Frontend::AgentTicketPriority###Subject

Description	Value
Description:	Sets the default subject for notes added in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Subject'} = '$Text{"Priority Update"}!';`

2.46.13. Ticket::Frontend::AgentTicketPriority###Body

Description	Value
Description:	Sets the default body text for notes added in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Body'} =  '';`

2.46.14. Ticket::Frontend::AgentTicketPriority###InvolvedAgent

Description	Value
Description:	Shows a list of all the involved agents on this ticket, in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'InvolvedAgent'} = '0';`

2.46.15. Ticket::Frontend::AgentTicketPriority###InformAgent

Description	Value
Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket priority screen of a zoomed ticket in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'InformAgent'} = '0';`

2.46.16. Ticket::Frontend::AgentTicketPriority###ArticleTypeDefault

Description	Value
Description:	Defines the default type of the note in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ArticleTypeDefault'} = 'note-internal';`

2.46.17. Ticket::Frontend::AgentTicketPriority###ArticleTypes

Description	Value
Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };`

2.46.18. Ticket::Frontend::AgentTicketPriority###Priority

Description	Value
Description:	Shows the ticket priority options in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0

Description	Value
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Priority'} = '1';`

2.46.19. Ticket::Frontend::AgentTicketPriority###PriorityDefault

Description	Value
Description:	Defines the default ticket priority in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'PriorityDefault'} = '3 normal';`

2.46.20. Ticket::Frontend::AgentTicketPriority###Title

Description	Value
Description:	Shows the title fields in the ticket priority screen of a zoomed ticket in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Title'} = '0';`

2.46.21. Ticket::Frontend::AgentTicketPriority###HistoryType

Description	Value
Description:	Defines the history type for the ticket priority screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'HistoryType'} = 'AddNote';`

2.46.22. Ticket::Frontend::AgentTicketPriority###HistoryComment

Description	Value
Description:	Defines the history comment for the ticket priority screen action, which gets used for ticket history in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'HistoryComment'} = '%%Priority';`

2.46.23. Ticket::Frontend::AgentTicketPriority###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket priority screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewPriority
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'DynamicField'} = {};`

2.47. Frontend::Agent::Ticket::ViewQueue

2.47.1. Ticket::Frontend::AgentTicketQueue###StripEmptyLines

Description	Value
Description:	Strips empty lines on the ticket preview in the queue view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'StripEmptyLines'} = '0';`

2.47.2. Ticket::Frontend::AgentTicketQueue###ViewAllPossibleTickets

Description	Value
Description:	Shows all both ro and rw queues in the queue view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'ViewAllPossibleTickets'} = '0';`

2.47.3. Ticket::Frontend::AgentTicketQueue###HighlightAge1

Description	Value
Description:	Sets the age in minutes (first level) for highlighting queues that contain untouched tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'HighlightAge1'} = '1440';`

2.47.4. Ticket::Frontend::AgentTicketQueue###HighlightAge2

Description	Value
Description:	Sets the age in minutes (second level) for highlighting queues that contain untouched tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'HighlightAge2'} = '2880';`

2.47.5. Ticket::Frontend::AgentTicketQueue###Blink

Description	Value
Description:	Activates a blinking mechanism of the queue that contains the oldest ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'Blink'} =  '1';`

2.47.6. Ticket::Frontend::AgentTicketQueue###QueueSort

Description	Value
Description:	Sorts the tickets (ascendingly or descendingly) when a single queue is selected in the queue view and after the tickets are sorted by priority. Values: 0 = ascending (oldest on top, default), 1 = descending (youngest on top). Use the QueueID for the key and 0 or 1 for value.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'QueueSort'} = { '3' => '0', '7' => '1' };`

2.47.7. Ticket::Frontend::AgentTicketQueue###SortBy::Default

Description	Value
Description:	Defines the default sort criteria for all queues displayed in the queue view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'SortBy::Default'} = 'Age';`

2.47.8. Ticket::Frontend::AgentTicketQueue###PreSort::ByPriority

Description	Value
Description:	Defines if a pre-sorting by priority should be done in the queue view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'PreSort::ByPriority'} = '1';`

2.47.9. Ticket::Frontend::AgentTicketQueue###Order::Default

Description	Value
Description:	Defines the default sort order for all queues in the queue view, after priority sort.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewQueue
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'Order::Default'} = 'Up';`

2.48. Frontend::Agent::Ticket::ViewResponsible

2.48.1. Ticket::Frontend::AgentTicketResponsible###Permission

Description	Value
Description:	Required permissions to use the ticket responsible screen in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Permission'} = 'responsible';`

2.48.2. Ticket::Frontend::AgentTicketResponsible###RequiredLock

Description	Value
Description:	Defines if a ticket lock is required in the ticket responsible screen of the agent interface (if the ticket isn't locked yet, the ticket gets locked and the current agent will be set automatically as its owner).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'RequiredLock'} = '0';`

2.48.3. Ticket::Frontend::AgentTicketResponsible###TicketType

Description	Value
Description:	Sets the ticket type in the ticket responsible screen of the agent interface (Ticket::Type needs to be activated).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'TicketType'} = '0';`

2.48.4. Ticket::Frontend::AgentTicketResponsible###Service

Description	Value
Description:	Sets the service in the ticket responsible screen of the agent interface (Ticket::Service needs to be activated).

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Service'} = '0';`

2.48.5. Ticket::Frontend::AgentTicketResponsible###Owner

Description	Value
Description:	Sets the ticket owner in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Owner'} = '0';`

2.48.6. Ticket::Frontend::AgentTicketResponsible###OwnerMandatory

Description	Value
Description:	Sets if ticket owner must be selected by the agent.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'OwnerMandatory'} = '0';`

2.48.7. Ticket::Frontend::AgentTicketResponsible###Responsible

Description	Value
Description:	Sets the responsible agent of the ticket in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Responsible'} = '1';`

2.48.8. Ticket::Frontend::AgentTicketResponsible###State

Description	Value
Description:	If a note is added by an agent, sets the state of a ticket in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'State'} = '0';`

2.48.9. Ticket::Frontend::AgentTicketResponsible###StateType

Description	Value
Description:	Defines the next state of a ticket after adding a note, in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'StateType'} = [ 'open', 'pending reminder', 'pending auto' ];`

2.48.10. Ticket::Frontend::AgentTicketResponsible###StateDefault

Description	Value
Description:	Defines the default next state of a ticket after adding a note, in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'StateDefault'} = 'open';`

2.48.11. Ticket::Frontend::AgentTicketResponsible###Note

Description	Value
Description:	Allows adding notes in the ticket responsible screen of the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Note'} = '1';`

2.48.12. Ticket::Frontend::AgentTicketResponsible###Subject

Description	Value
Description:	Sets the default subject for notes added in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Subject'} = '$Text{"Responsible Update"}!';`

2.48.13. Ticket::Frontend::AgentTicketResponsible###Body

Description	Value
Description:	Sets the default body text for notes added in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Body'} = '';`

2.48.14. Ticket::Frontend::AgentTicketResponsible###InvolvedAgent

Description	Value
Description:	Shows a list of all the involved agents on this ticket, in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'InvolvedAgent'} = '0';`

2.48.15. Ticket::Frontend::AgentTicketResponsible###InformAgent

Description	Value
Description:	Shows a list of all the possible agents (all agents with note permissions on the queue/ticket) to determine who should be informed about this note, in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'InformAgent'} = '0';`

2.48.16. Ticket::Frontend::AgentTicketResponsible###ArticleTypeDefault

Description	Value
Description:	Defines the default type of the note in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'ArticleTypeDefault'} = 'note-internal';`

2.48.17. Ticket::Frontend::AgentTicketResponsible###ArticleTypes

Description	Value
Description:	Specifies the different note types that will be used in the system.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'ArticleTypes'} = { 'note-external' => '0', 'note-internal' => '1', 'note-report' => '0' };`

2.48.18. Ticket::Frontend::AgentTicketResponsible###Priority

Description	Value
Description:	Shows the ticket priority options in the ticket responsible screen of the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Priority'} = '0';`

2.48.19. Ticket::Frontend::AgentTicketResponsible###PriorityDefault

Description	Value
Description:	Defines the default ticket priority in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'PriorityDefault'} = '3 normal';`

2.48.20. Ticket::Frontend::AgentTicketResponsible###Title

Description	Value
Description:	Shows the title fields in the ticket responsible screen of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Title'} = '1';`

2.48.21. Ticket::Frontend::AgentTicketResponsible###HistoryType

Description	Value
Description:	Defines the history type for the ticket responsible screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'HistoryType'} = 'AddNote';`

2.48.22. Ticket::Frontend::AgentTicketResponsible###HistoryComment

Description	Value
Description:	Defines the history comment for the ticket responsible screen action, which gets used for ticket history in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'HistoryComment'} = '%%Responsible';`

2.48.23. Ticket::Frontend::AgentTicketResponsible###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket responsible screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewResponsible
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'DynamicField'} = {};`

2.49. Frontend::Agent::Ticket::ViewSearch

2.49.1. Ticket::Frontend::AgentTicketSearch###ExtendedSearchCondition

Description	Value
Description:	Allows extended search conditions in ticket search of the agent interface. With this feature you can search e. g. with this kind of conditions like "(key1&&key2)" or "(key1||key2)".
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'ExtendedSearchCondition'} = '1';`

2.49.2. Ticket::Frontend::AgentTicketSearch###SearchLimit

Description	Value
Description:	Maximum number of tickets to be displayed in the result of a search in the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchLimit'} = '2000';`

2.49.3. Ticket::Frontend::AgentTicketSearch###SearchPageShown

Description	Value
Description:	Number of tickets to be displayed in each page of a search result in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchPageShown'} = '40';`

2.49.4. Ticket::Frontend::AgentTicketSearch###SearchViewableTicketLines

Description	Value
Description:	Number of lines (per ticket) that are shown by the search utility in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchViewableTicketLines'} = '10';`

2.49.5. Ticket::Frontend::AgentTicketSearch###SortBy::Default

Description	Value
Description:	Defines the default ticket attribute for ticket sorting of the ticket search result of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SortBy::Default'} = 'Age';`

2.49.6. Ticket::Frontend::AgentTicketSearch###Order::Default

Description	Value
Description:	Defines the default ticket order in the ticket search result of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Order::Default'} = 'Down';`

2.49.7. Ticket::Frontend::AgentTicketSearch###SearchArticleCSVTree

Description	Value
Description:	Exports the whole article tree in search result (it can affect the system performance).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchArticleCSVTree'} = '0';`

2.49.8. Ticket::Frontend::AgentTicketSearch###SearchCSVData

Description	Value
Description:	Data used to export the search result in CSV format.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchCSVData'} =
[
 'TicketNumber',
 'Age',
 'Created',
 'Closed',
 'FirstLock',
 'FirstResponse',
 'State',
 'Priority',
 'Queue',
 'Lock',
 'Owner',
 'UserFirstname',
 'UserLastname',
```

Description	Value

	`'CustomerID',`
	`'CustomerName',`
	`'From',`
	`'Subject',`
	`'AccountedTime',`
	`'ArticleTree',`
	`'SolutionInMin',`
	`'SolutionDiffInMin',`
	`'FirstResponseInMin',`
	`'FirstResponseDiffInMin'`
	`];`

2.49.9. Ticket::Frontend::AgentTicketSearch###ArticleCreateTime

Description	Value
Description:	Includes article create times in the ticket search of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'ArticleCreateTime'} = '0';`

2.49.10. Ticket::Frontend::AgentTicketSearch###Defaults###Fulltext

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Fulltext'} = '';`

2.49.11. Ticket::Frontend::AgentTicketSearch###Defaults###TicketNumber

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketNumber'} = '';`

2.49.12. Ticket::Frontend::AgentTicketSearch###Defaults###Title

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Title'} = '';`

2.49.13. Ticket::Frontend::AgentTicketSearch###Defaults###From

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'From'} = '';`

2.49.14. Ticket::Frontend::AgentTicketSearch###Defaults###To

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'To'} = '';`

2.49.15. Ticket::Frontend::AgentTicketSearch###Defaults###Cc

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Cc'} = '';`

2.49.16. Ticket::Frontend::AgentTicketSearch###Defaults###Subject

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Subject'} = '';`

2.49.17. Ticket::Frontend::AgentTicketSearch###Defaults###Body

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Body'} = '';`

2.49.18. Ticket::Frontend::AgentTicketSearch###Defaults###CustomerID

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	

Description	Value
	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'CustomerID'} = '';`

2.49.19. Ticket::Frontend::AgentTicketSearch###Defaults###CustomerUserLogin

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'CustomerUserLogin'} = '';`

2.49.20. Ticket::Frontend::AgentTicketSearch###Defaults###StateIDs

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'StateIDs'} = [];`

2.49.21. Ticket::Frontend::AgentTicketSearch###Defaults###QueueIDs

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'QueueIDs'} = [];`

2.49.22. Ticket::Frontend::AgentTicketSearch###Defaults###TicketCreateTimePoint

Description	Value
Description:	Default data to use on attribute for ticket search screen. Example: "TicketCreateTimePointFormat=year;TicketCreateTimePointStart=Last;TicketCreate

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCreateTimePoint'} = '';`

2.49.23. Ticket::Frontend::AgentTicketSearch###Defaults###TicketCreateTimeSlot

Description	Value
Description:	Default data to use on attribute for ticket search screen. Example: "TicketCreateTimeStartYear=2010;TicketCreateTimeStartMonth=10;TicketCreateTi
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCreateTimeSlot'} = '';`

2.49.24. Ticket::Frontend::AgentTicketSearch###Defaults###TicketChangeTimePoint

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketChangeTimePoint'} = '';`

2.49.25. Ticket::Frontend::AgentTicketSearch###Defaults###TicketChangeTimeSlot

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0

Description	Value
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketChangeTimeSlot'} = '';`

2.49.26. Ticket::Frontend::AgentTicketSearch###Defaults###TicketCloseTimePoint

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCloseTimePoint'} = '';`

2.49.27. Ticket::Frontend::AgentTicketSearch###Defaults###TicketCloseTimeSlot

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCloseTimeSlot'} = '';`

2.49.28. Ticket::Frontend::AgentTicketSearch###Defaults###ArticleCreateTimePoint

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'ArticleCreateTimePoint'} = '';`

2.49.29. Ticket::Frontend::AgentTicketSearch###Defaults###ArticleCreateTimeSlot

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'ArticleCreateTimeSlot'} = '';`

2.49.30. Ticket::Frontend::AgentTicketSearch###Defaults###SearchInArchive

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'SearchInArchive'} = '';`

2.49.31. Ticket::Frontend::CustomerTicketSearch###SearchArticleCSVTree

Description	Value
Description:	Exports the whole article tree in search result (it can affect the system performance).
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchArticleCSVTree'} = '0';`

2.49.32. Ticket::Frontend::AgentTicketSearch###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket search screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1

Description	Value
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'DynamicField'} = {};`

2.49.33. Ticket::Frontend::AgentTicketSearch###Defaults###DynamicField

Description	Value
Description:	Defines the default shown ticket search attribute for ticket search screen. Example: a text, 1, Search_DynamicField_Field1StartYear=2002; Search_DynamicField_Field1StartMonth=12; Search_DynamicField_Field1StartDay=12; Search_DynamicField_Field1StartHour=00; Search_DynamicField_Field1StartMinute=00; Search_DynamicField_Field1StartSecond=00; Search_DynamicField_Field1StopYear=2009; Search_DynamicField_Field1StopMonth=02; Search_DynamicField_Field1StopDay=10; Search_DynamicField_Field1StopHour=23; Search_DynamicField_Field1StopMinute=59; Search_DynamicField_Field1StopSecond=59;.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'DynamicField'} = {};`

2.49.34. Ticket::Frontend::AgentTicketSearch###SearchCSVDynamicField

Description	Value
Description:	Dynamic Fields used to export the search result in CSV format.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchCSVDynamicField'} = {};`

2.50. Frontend::Agent::Ticket::ViewStatus

2.50.1. Ticket::Frontend::AgentTicketStatusView###ViewableTicketsPage

Description	Value
Description:	Shows all open tickets (even if they are locked) in the status view of the agent interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'ViewableTicketsPage'} = '50';`

2.50.2. Ticket::Frontend::AgentTicketStatusView###SortBy::Default

Description	Value
Description:	Defines the default ticket attribute for ticket sorting in the status view of the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'SortBy::Default'} = 'Age';`

2.50.3. Ticket::Frontend::AgentTicketStatusView###Order::Default

Description	Value
Description:	Defines the default ticket order (after priority sort) in the status view of the agent interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewStatus
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'Order::Default'} = 'Down';`

2.51. Frontend::Agent::Ticket::ViewZoom

2.51.1. Ticket::Frontend::PlainView

Description	Value
Description:	Shows a link to see a zoomed email ticket in plain text.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'Ticket::Frontend::PlainView'} = '0';`

2.51.2. Ticket::Frontend::ZoomExpand

Description	Value
Description:	Shows all the articles of the ticket (expanded) in the zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ZoomExpand'} = '0';`

2.51.3. Ticket::Frontend::ZoomExpandSort

Description	Value
Description:	Shows the articles sorted normally or in reverse, under ticket zoom in the agent interface.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ZoomExpandSort'} = 'normal';`

2.51.4. Ticket::ZoomAttachmentDisplayCount

Description	Value
Description:	Shows a count of icons in the ticket zoom, if the article has attachments.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::ZoomAttachmentDisplayCount'} = '20';`

2.51.5. Ticket::ZoomTimeDisplay

Description	Value
Description:	Displays the accounted time for an article in the ticket zoom view.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	0

Description	Value
Required:	0
Config-Setting:	`$Self->{'Ticket::ZoomTimeDisplay'} = '0';`

2.51.6. Ticket::UseArticleColors

Description	Value
Description:	Shows colors for different article types in the article table.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::UseArticleColors'} = '0';`

2.51.7. Ticket::Frontend::TicketArticleFilter

Description	Value
Description:	Activates the article filter in the zoom view to specify which articles should be shown.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::TicketArticleFilter'} = '0';`

2.51.8. Ticket::Frontend::HTMLArticleHeightDefault

Description	Value
Description:	Set the default height (in pixels) of inline HTML articles in AgentTicketZoom.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::HTMLArticleHeightDefault'} = '100';`

2.51.9. Ticket::Frontend::HTMLArticleHeightMax

Description	Value
Description:	Set the maximum height (in pixels) of inline HTML articles in AgentTicketZoom.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::HTMLArticleHeightMax'} = '2500';`

2.51.10. Ticket::Frontend::ZoomRichTextForce

Description	Value
Description:	Show article as rich text even if rich text writing is disabled.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::ZoomRichTextForce'} = '0';`

2.51.11. Ticket::Frontend::AgentTicketZoom###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket zoom screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Agent::Ticket::ViewZoom
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::AgentTicketZoom'}->{'DynamicField'} = {};`

2.52. Frontend::Agent::TicketOverview

2.52.1. Ticket::Frontend::Overview###Small

Description	Value
Description:	Allows having a small format ticket overview (CustomerInfo => 1 - shows also the customer information).
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::Overview'}->{'Small'} = { 'CustomerInfo' => '1', 'Module' => 'Kernel::Output::HTML::TicketOverviewSmall', 'Name' => 'Small', 'NameShort' => 'S'`

Description	Value
	`};`

2.52.2. Ticket::Frontend::OverviewSmall###ColumnHeader

Description	Value
Description:	Shows either the last customer article's subject or the ticket title in the small format overview.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::OverviewSmall'}->{'ColumnHeader'} = 'LastCustomerSubject';`

2.52.3. Ticket::Frontend::Overview###Medium

Description	Value
Description:	Allows having a medium format ticket overview (CustomerInfo => 1 - shows also the customer information).
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::Overview'}->{'Medium'} = { 'CustomerInfo' => '0', 'Module' => 'Kernel::Output::HTML::TicketOverviewMedium', 'Name' => 'Medium', 'NameShort' => 'M', 'TicketActionsPerTicket' => '0' };`

2.52.4. Ticket::Frontend::Overview###Preview

Description	Value
Description:	Shows a preview of the ticket overview (CustomerInfo => 1 - shows also Customer-Info, CustomerInfoMaxSize max. size in characters of Customer-Info).
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::Overview'}->{'Preview'} = { 'CustomerInfo' => '0', 'CustomerInfoMaxSize' => '18', 'DefaultPreViewLines' => '25',`

Description	Value

	`'DefaultViewNewLine' => '90',`
	`'Module' => 'Kernel::Output::HTML::TicketOverviewPreview',`
	`'Name' => 'Preview',`
	`'NameShort' => 'L',`
	`'StripEmptyLines' => '0',`
	`'TicketActionsPerTicket' => '0'`
	`};`

2.52.5. Ticket::Frontend::Overview::PreviewArticleSenderTypes

Description	Value
Description:	Defines which article sender types should be shown in the preview of a ticket.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Ticket::Frontend::Overview::PreviewArticleSenderTypes'} = {`
	`'agent' => '1',`
	`'customer' => '1',`
	`'system' => '1'`
	`};`

2.52.6. Ticket::Frontend::Overview::PreviewArticleTypeExpanded

Description	Value
Description:	Defines wich article type should be expanded when entering the overview. If nothing defined, latest article will be expanded.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::Overview::PreviewArticleTypeExpanded'} = '';`

2.52.7. Ticket::Frontend::OverviewSmall###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket small format overview screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::OverviewSmall'}->{'DynamicField'} = {};`

2.52.8. Ticket::Frontend::OverviewMedium###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket medium format overview screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::OverviewMedium'}->{'DynamicField'} = {};`

2.52.9. Ticket::Frontend::OverviewPreview###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket preview format overview screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Agent::TicketOverview
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::OverviewPreview'}->{'DynamicField'} = {};`

2.53. Frontend::Agent::ToolBarModule

2.53.1. Frontend::ToolBarModule###1-Ticket::AgentTicketQueue

Description	Value
Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0
Config-Setting:	

```
$Self->{'Frontend::ToolBarModule'}->{'1-Ticket::AgentTicketQueue'} =
{
 'AccessKey' => 'q',
 'Action' => 'AgentTicketQueue',
 'CssClass' => 'QueueView',
 'Link' => 'Action=AgentTicketQueue',
 'Module' => 'Kernel::Output::HTML::ToolBarLink',
 'Name' => 'Queue view',
 'Priority' => '1010010'
};
```

2.53.2. Frontend::ToolBarModule###2-Ticket::AgentTicketStatus

Description	Value
Description:	Toolbar Item for a shortcut.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'2-Ticket::AgentTicketStatus'} = { 'AccessKey' => 'o', 'Action' => 'AgentTicketStatusView', 'CssClass' => 'StatusView', 'Link' => 'Action=AgentTicketStatusView', 'Module' => 'Kernel::Output::HTML::ToolBarLink', 'Name' => 'Status view', 'Priority' => '1010020' };`

2.53.3. Frontend::ToolBarModule###3-Ticket::AgentTicketEscalation

Description	Value
Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'3-Ticket::AgentTicketEscalation'} = { 'AccessKey' => 'w', 'Action' => 'AgentTicketEscalationView', 'CssClass' => 'EscalationView', 'Link' => 'Action=AgentTicketEscalationView', 'Module' => 'Kernel::Output::HTML::ToolBarLink', 'Name' => 'Escalation view', 'Priority' => '1010030' };`

2.53.4. Frontend::ToolBarModule###4-Ticket::AgentTicketPhone

Description	Value
Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'4-Ticket::AgentTicketPhone'} = { 'AccessKey' => 'l', 'Action' => 'AgentTicketPhone', 'CssClass' => 'PhoneTicket', 'Link' => 'Action=AgentTicketPhone', 'Module' => 'Kernel::Output::HTML::ToolBarLink',`

Description	Value

	`'Name' => 'New phone ticket',`
	`'Priority' => '1020010'`
	`};`

2.53.5. Frontend::ToolBarModule###5-Ticket::AgentTicketEmail

Description	Value
Description:	Toolbar Item for a shortcut.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'5-Ticket::AgentTicketEmail'} =`
	`{`
	`'AccessKey' => 'l',`
	`'Action' => 'AgentTicketEmail',`
	`'CssClass' => 'EmailTicket',`
	`'Link' => 'Action=AgentTicketEmail',`
	`'Module' => 'Kernel::Output::HTML::ToolBarLink',`
	`'Name' => 'New email ticket',`
	`'Priority' => '1020020'`
	`};`

2.53.6. Frontend::ToolBarModule###6-Ticket::TicketResponsible

Description	Value
Description:	Agent interface notification module to see the number of tickets an agent is responsible for.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	1
Required:	0

Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'6-Ticket::TicketResponsible'} =`
	`{`
	`'CssClass' => 'Responsible',`
	`'CssClassNew' => 'Responsible New',`
	`'CssClassReached' => 'Responsible Reached',`
	`'Module' => 'Kernel::Output::HTML::ToolBarTicketResponsible',`
	`'Priority' => '1030010'`
	`};`

2.53.7. Frontend::ToolBarModule###7-Ticket::TicketWatcher

Description	Value
Description:	Agent interface notification module to see the number of watched tickets.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	1

Description	Value
Required:	0
Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'7-Ticket::TicketWatcher'} = { 'CssClass' => 'Watcher', 'CssClassNew' => 'Watcher New', 'CssClassReached' => 'Watcher Reached', 'Module' => 'Kernel::Output::HTML::ToolBarTicketWatcher', 'Priority' => '1030020' };`

2.53.8. Frontend::ToolBarModule###8-Ticket::TicketLocked

Description	Value
Description:	Agent interface notification module to check the used charset.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'8-Ticket::TicketLocked'} = { 'CssClass' => 'Locked', 'CssClassNew' => 'Locked New', 'CssClassReached' => 'Locked Reached', 'Module' => 'Kernel::Output::HTML::ToolBarTicketLocked', 'Priority' => '1030030' };`

2.53.9. Frontend::ToolBarModule###9-Ticket::TicketSearchProfile

Description	Value
Description:	Agent interface module to access search profiles via nav bar.
Group:	Ticket
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'9-Ticket::TicketSearchProfile'} = { 'Block' => 'ToolBarSearchProfile', 'Description' => 'Search-Template', 'MaxWidth' => '40', 'Module' => 'Kernel::Output::HTML::ToolBarTicketSearchProfile', 'Name' => 'Search-Template', 'Priority' => '1990010' };`

2.53.10. Frontend::ToolBarModule###10-Ticket::TicketSearchFulltext

Description	Value
Description:	Agent interface module to access fulltext search via nav bar.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Agent::ToolBarModule
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::ToolBarModule'}->{'10-Ticket::TicketSearchFulltext'} = { 'Block' => 'ToolBarSearchFulltext', 'CSS' => 'Core.Agent.Toolbar.FulltextSearch.css', 'Description' => 'Fulltext-Search', 'Module' => 'Kernel::Output::HTML::ToolBarTicketSearchFulltext', 'Name' => 'Fulltext-Search', 'Priority' => '1990020', 'Size' => '10' };`

2.54. Frontend::Customer

2.54.1. Ticket::Frontend::CustomerTicketOverviewSortable

Description	Value
Description:	Controls if customers have the ability to sort their tickets.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketOverviewSortable'} = '';`

2.54.2. Ticket::Frontend::CustomerTicketOverviewCustomEmptyText

Description	Value
Description:	Custom text for the page shown to customers that have no tickets yet.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketOverviewCustomEmptyText'} = { 'Button' => 'Create your first ticket', 'Text' => 'Please click the button below to create your first ticket.', 'Title' => 'Welcome!' };`

2.54.3. Ticket::Frontend::CustomerTicketOverview###ColumnHeader

Description	Value
Description:	Shows either the last customer article's subject or the ticket title in the small format overview.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketOverview'}->{'ColumnHeader'} = 'TicketTitle';`

2.54.4. Frontend::CustomerUser::Item###15-OpenTickets

Description	Value
Description:	Customer item (icon) which shows the open tickets of this customer as info block. Setting CustomerUserLogin to 1 searches for tickets based on login name rather than CustomerID.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Frontend::CustomerUser::Item'}->{'15-OpenTickets'} = { 'Action' => 'AgentTicketSearch', 'Attributes' => 'StateType=Open;', 'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css', 'CSSClassNoOpenTicket' => 'NoOpenTicket', 'CSSClassOpenTicket' => 'OpenTicket', 'CustomerUserLogin' => '0', 'Module' => 'Kernel::Output::HTML::CustomerUserGenericTicket', 'Subaction' => 'Search', 'Target' => '_blank', 'Text' => 'Open tickets' };`

2.54.5. Frontend::CustomerUser::Item###16-OpenTicketsForCustomerUserLogin

Description	Value
Description:	Customer item (icon) which shows the open tickets of this customer as info block. Setting CustomerUserLogin to 1 searches for tickets based on login name rather than CustomerID.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Frontend::CustomerUser::Item'}->{'16-OpenTicketsForCustomerUserLogin'} = { 'Action' => 'AgentTicketSearch', 'Attributes' => 'StateType=Open;', 'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css', 'CSSClassNoOpenTicket' => 'NoOpenTicket', 'CSSClassOpenTicket' => 'OpenTicket',`

Description	Value

	`'CustomerUserLogin' => '1',`
	`'Module' => 'Kernel::Output::HTML::CustomerUserGenericTicket',`
	`'Subaction' => 'Search',`
	`'Target' => '_blank',`
	`'Text' => 'Open tickets of customer'`
	`};`

2.54.6. Frontend::CustomerUser::Item###17-ClosedTickets

Description	Value
Description:	Customer item (icon) which shows the closed tickets of this customer as info block. Setting CustomerUserLogin to 1 searches for tickets based on login name rather than CustomerID.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::CustomerUser::Item'}->{'17-ClosedTickets'} = {`
	`'Action' => 'AgentTicketSearch',`
	`'Attributes' => 'StateType=Closed;',`
	`'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css',`
	`'CSSClassNoOpenTicket' => 'NoOpenTicket',`
	`'CSSClassOpenTicket' => 'OpenTicket',`
	`'CustomerUserLogin' => '0',`
	`'Module' => 'Kernel::Output::HTML::CustomerUserGenericTicket',`
	`'Subaction' => 'Search',`
	`'Target' => '_blank',`
	`'Text' => 'Closed tickets'`
	`};`

2.54.7. Frontend::CustomerUser::Item###18-ClosedTicketsForCustomerUserLogin

Description	Value
Description:	Customer item (icon) which shows the closed tickets of this customer as info block. Setting CustomerUserLogin to 1 searches for tickets based on login name rather than CustomerID.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	0
Required:	0

Config-Setting:	`$Self->{'Frontend::CustomerUser::Item'}->{'18-`
	`ClosedTicketsForCustomerUserLogin'} = {`
	`'Action' => 'AgentTicketSearch',`
	`'Attributes' => 'StateType=Closed;',`
	`'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css',`
	`'CSSClassNoOpenTicket' => 'NoOpenTicket',`
	`'CSSClassOpenTicket' => 'OpenTicket',`
	`'CustomerUserLogin' => '1',`
	`'Module' => 'Kernel::Output::HTML::CustomerUserGenericTicket',`
	`'Subaction' => 'Search',`
	`'Target' => '_blank',`
	`'Text' => 'Closed tickets of customer'`

Description	Value
	`};`

2.54.8. CustomerFrontend::CommonObject###QueueObject

Description	Value
Description:	Path of the file that stores all the settings for the QueueObject object for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerFrontend::CommonObject'}->{'QueueObject'} = 'Kernel::System::Queue';`

2.54.9. CustomerFrontend::CommonObject###TicketObject

Description	Value
Description:	Path of the file that stores all the settings for the TicketObject for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerFrontend::CommonObject'}->{'TicketObject'} = 'Kernel::System::Ticket';`

2.54.10. CustomerFrontend::CommonParam###Action

Description	Value
Description:	Defines the default used Frontend-Module if no Action parameter given in the url on the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerFrontend::CommonParam'}->{'Action'} = 'CustomerTicketOverview';`

2.54.11. CustomerFrontend::CommonParam###TicketID

Description	Value
Description:	Default ticket ID used by the system in the customer interface.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Customer
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerFrontend::CommonParam'}->{'TicketID'} = '';`

2.55. Frontend::Customer::ModuleMetaHead

2.55.1. CustomerFrontend::HeaderMetaModule###2-TicketSearch

Description	Value
Description:	Module to generate html OpenSearch profile for short ticket search in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleMetaHead
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerFrontend::HeaderMetaModule'}->{'2-TicketSearch'} = { 'Action' => 'CustomerTicketSearch', 'Module' => 'Kernel::Output::HTML::CustomerHeaderMetaTicketSearch' };`

2.56. Frontend::Customer::ModuleRegistration

2.56.1. CustomerFrontend::Module###CustomerTicketOverview

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	

```
$Self->{'CustomerFrontend::Module'}->{'CustomerTicketOverview'} =  {
 'Description' => 'Overview of customer tickets',
 'NavBar' => [
   {
     'AccessKey' => 'm',
     'Block' => '',
     'Description' => 'My Tickets',
     'Link' => 'Action=CustomerTicketOverview;Subaction=MyTickets',
     'LinkOption' => '',
     'Name' => 'My Tickets',
     'NavBar' => '',
     'Prio' => '110',
     'Type' => ''
   },
   {
     'AccessKey' => 'c',
     'Block' => '',
     'Description' => 'Company Tickets',
```

Description	Value
	`'Link' => 'Action=CustomerTicketOverview;Subaction=CompanyTickets', 'LinkOption' => '', 'Name' => 'Company Tickets', 'NavBar' => '', 'Prio' => '120', 'Type' => '' } ], 'NavBarName' => 'Ticket', 'Title' => 'Overview' };`

2.56.2. CustomerFrontend::Module###CustomerTicketMessage

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerTicketMessage'} = { 'Description' => 'Create tickets', 'NavBar' => [{ 'AccessKey' => 'n', 'Block' => '', 'Description' => 'Create new Ticket', 'Link' => 'Action=CustomerTicketMessage', 'LinkOption' => '', 'Name' => 'New Ticket', 'NavBar' => '', 'Prio' => '100', 'Type' => '' }], 'NavBarName' => 'Ticket', 'Title' => 'New Ticket' };`

2.56.3. CustomerFrontend::Module###CustomerTicketZoom

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerTicketZoom'} = { 'Description' => 'Ticket zoom view', 'Loader' => { 'JavaScript' => [ 'Core.Customer.TicketZoom.js',`

Description	Value

	`'Core.UI.Popup.js'`
	`]`
	`},`
	`'NavBarName' => 'Ticket',`
	`'Title' => 'Zoom'`
	`};`

2.56.4. CustomerFrontend::Module###CustomerTicketPrint

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

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerTicketPrint'} = {`
	`'Description' => 'Customer Ticket Print Module',`
	`'NavBarName' => '',`
	`'Title' => 'Print'`
	`};`

2.56.5. CustomerFrontend::Module###CustomerTicketAttachment

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

Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerTicketAttachment'} =`
	`{`
	`'Description' => 'To download attachments',`
	`'NavBarName' => '',`
	`'Title' => ''`
	`};`

2.56.6. CustomerFrontend::Module###CustomerTicketSearch

Description	Value
Description:	Frontend module registration for the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::ModuleRegistration
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerFrontend::Module'}->{'CustomerTicketSearch'} = {` `'Description' => 'Customer ticket search',`

Description	Value

	``
	  'NavBar' => [
	    {
	      'AccessKey' => 's',
	      'Block' => '',
	      'Description' => 'Search',
	      'Link' => 'Action=CustomerTicketSearch',
	      'LinkOption' => '',
	      'Name' => 'Search',
	      'NavBar' => '',
	      'Prio' => '300',
	      'Type' => ''
	    }
	  ],
	  'NavBarName' => 'Ticket',
	  'Title' => 'Search'
	};
	``

2.57. Frontend::Customer::Preferences

2.57.1. CustomerPreferencesGroups###ShownTickets

Description	Value
Description:	Defines all the parameters for the ShownTickets object in the customer preferences of the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0

Config-Setting:	```
	$Self->{'CustomerPreferencesGroups'}->{'ShownTickets'} = {
	  'Active' => '1',
	  'Column' => 'User Profile',
	  'Data' => {
	    '15' => '15',
	    '20' => '20',
	    '25' => '25',
	    '30' => '30'
	  },
	  'DataSelected' => '25',
	  'Key' => 'Tickets per page',
	  'Label' => 'Number of displayed tickets',
	  'Module' => 'Kernel::Output::HTML::PreferencesGeneric',
	  'PrefKey' => 'UserShowTickets',
	  'Prio' => '4000'
	};
	``

2.57.2. CustomerPreferencesGroups###RefreshTime

Description	Value
Description:	Defines all the parameters for the RefreshTime object in the customer preferences of the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Preferences
Valid:	1
Required:	0

Description	Value
Config-Setting:	`$Self->{'CustomerPreferencesGroups'}->{'RefreshTime'} = { 'Active' => '1', 'Column' => 'User Profile', 'Data' => { '' => 'off', '10' => '10 minutes', '15' => '15 minutes', '2' => ' 2 minutes', '5' => ' 5 minutes', '7' => ' 7 minutes' }, 'DataSelected' => '', 'Key' => 'Refresh interval', 'Label' => 'Ticket overview', 'Module' => 'Kernel::Output::HTML::PreferencesGeneric', 'PrefKey' => 'UserRefreshTime', 'Prio' => '4000' };`

2.58. Frontend::Customer::Ticket::ViewNew

2.58.1. Ticket::Frontend::CustomerTicketMessage###NextScreenAfterNewTicket

Description	Value
Description:	Determines the next screen after new customer ticket in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'NextScreenAfterNewTicket'} = 'CustomerTicketOverview';`

2.58.2. Ticket::Frontend::CustomerTicketMessage###Priority

Description	Value
Description:	Allows customers to set the ticket priority in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'Priority'} = '1';`

2.58.3. Ticket::Frontend::CustomerTicketMessage###PriorityDefault

Description	Value
Description:	Defines the default priority of new customer tickets in the customer interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'PriorityDefault'} = '3 normal';`

2.58.4. Ticket::Frontend::CustomerTicketMessage###Queue

Description	Value
Description:	Allows customers to set the ticket queue in the customer interface. If this is set to 'No', QueueDefault should be configured.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'Queue'} = '1';`

2.58.5. Ticket::Frontend::CustomerTicketMessage###QueueDefault

Description	Value
Description:	Defines the default queue for new customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	0
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'QueueDefault'} = 'Postmaster';`

2.58.6. Ticket::Frontend::CustomerTicketMessage###Service

Description	Value
Description:	Allows customers to set the ticket service in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'Service'} = '1';`

2.58.7. Ticket::Frontend::CustomerTicketMessage###SLA

Description	Value
Description:	Allows customers to set the ticket SLA in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'SLA'} = '1';`

2.58.8. Ticket::Frontend::CustomerTicketMessage###StateDefault

Description	Value
Description:	Defines the default state of new customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'StateDefault'} = 'new';`

2.58.9. Ticket::Frontend::CustomerTicketMessage###ArticleType

Description	Value
Description:	Defines the default type for article in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'ArticleType'} = 'webrequest';`

2.58.10. Ticket::Frontend::CustomerTicketMessage###SenderType

Description	Value
Description:	Sender type for new tickets from the customer inteface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'SenderType'} = 'customer';`

2.58.11. Ticket::Frontend::CustomerTicketMessage###HistoryType

Description	Value
Description:	Defines the default history type in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'HistoryType'} = 'WebRequestCustomer';`

2.58.12. Ticket::Frontend::CustomerTicketMessage###HistoryComment

Description	Value
Description:	Comment for new history entries in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'HistoryComment'} = '';`

2.58.13. CustomerPanelSelectionType

Description	Value
Description:	Defines the receipent target of the tickets ("Queue" shows all queues, "SystemAddress" displays all system addresses) in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanelSelectionType'} = 'Queue';`

2.58.14. CustomerPanelSelectionString

Description	Value
Description:	Determines the strings that will be shown as receipent (To:) of the ticket in the customer interface. For Queue as CustomerPanelSelectionType, "<Queue>" shows the names of the

Description	Value
	queues, and for SystemAddress, "<Realname> <<Email>>" shows the name and email of the receipent.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'CustomerPanelSelectionString'} = '<Queue>';`

2.58.15. CustomerPanelOwnSelection

Description	Value
Description:	Determines which queues will be valid for ticket's recepients in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	0
Required:	0
Config-Setting:	`$Self->{'CustomerPanelOwnSelection'} = { 'Junk' => 'First Queue', 'Misc' => 'Second Queue' };`

2.58.16. CustomerPanel::NewTicketQueueSelectionModule

Description	Value
Description:	Module for To-selection in new ticket screen in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	1
Config-Setting:	`$Self->{'CustomerPanel::NewTicketQueueSelectionModule'} = 'Kernel::Output::HTML::CustomerNewTicketQueueSelectionGeneric';`

2.58.17. Ticket::Frontend::CustomerTicketMessage###DynamicField

Description	Value
Description:	Dynamic fields options shown in the ticket message screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required. NOTE. If you want to display these fields also in the ticket zoom of the customer interface, you have to enable them in CustomerTicketZoom###DynamicField.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Customer::Ticket::ViewNew
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'DynamicField'} = {};`

2.59. Frontend::Customer::Ticket::ViewPrint

2.59.1. Ticket::Frontend::CustomerTicketPrint###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket print screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewPrint
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketPrint'}->{'DynamicField'} = {};`

2.60. Frontend::Customer::Ticket::ViewSearch

2.60.1. Ticket::CustomerTicketSearch::SearchLimit

Description	Value
Description:	Maximum number of tickets to be displayed in the result of a search in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomerTicketSearch::SearchLimit'} = '5000';`

2.60.2. Ticket::CustomerTicketSearch::SearchPageShown

Description	Value
Description:	Number of tickets to be displayed in each page of a search result in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'Ticket::CustomerTicketSearch::SearchPageShown'} =  '40';`

2.60.3. Ticket::CustomerTicketSearch::SortBy::Default

Description	Value
Description:	Defines the default ticket attribute for ticket sorting in a ticket search of the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomerTicketSearch::SortBy::Default'} =  'Age';`

2.60.4. Ticket::CustomerTicketSearch::Order::Default

Description	Value
Description:	Defines the default ticket order of a search result in the customer interface. Up: oldest on top. Down: latest on top.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::CustomerTicketSearch::Order::Default'} =  'Down';`

2.60.5. Ticket::Frontend::CustomerTicketSearch###ExtendedSearchCondition

Description	Value
Description:	Allows extended search conditions in ticket search of the customer interface. With this feature you can search w. g. with this kind of conditions like "(key1&&key2)" or "(key1||key2)".
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'ExtendedSearchCondition'} =  '1';`

2.60.6. Ticket::Frontend::CustomerTicketSearch###SearchCSVData

Description	Value
Description:	Data used to export the search result in CSV format.
Group:	Ticket

Description	Value
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchCSVData'} = [ 'TicketNumber', 'Age', 'Created', 'Closed', 'State', 'Priority', 'Lock', 'CustomerID', 'CustomerName', 'From', 'Subject' ];`

2.60.7. Ticket::Frontend::CustomerTicketSearch###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket search screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'DynamicField'} = {};`

2.60.8. Ticket::Frontend::CustomerTicketSearch###SearchOverviewDynamicField

Description	Value
Description:	Dynamic fields shown in the ticket search overview results screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchOverviewDynamicField'} = {};`

2.60.9. Ticket::Frontend::CustomerTicketSearch###SearchCSVDynamicField

Description	Value
Description:	Dynamic Fields used to export the search result in CSV format.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewSearch
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchCSVDynamicField'} = {};`

2.61. Frontend::Customer::Ticket::ViewZoom

2.61.1. Ticket::Frontend::CustomerTicketZoom###NextScreenAfterFollowUp

Description	Value
Description:	Determines the next screen after the follow up screen of a zoomed ticket in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'NextScreenAfterFollowUp'} = 'CustomerTicketOverview';`

2.61.2. Ticket::Frontend::CustomerTicketZoom###ArticleType

Description	Value
Description:	Defines the default type of the note in the ticket zoom screen of the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'ArticleType'} = 'webrequest';`

2.61.3. Ticket::Frontend::CustomerTicketZoom###SenderType

Description	Value
Description:	Defines the default sender type for tickets in the ticket zoom screen of the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	

Description	Value
	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'SenderType'} = 'customer';`

2.61.4. Ticket::Frontend::CustomerTicketZoom###HistoryType

Description	Value
Description:	Defines the history type for the ticket zoom action, which gets used for ticket history in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'HistoryType'} = 'FollowUp';`

2.61.5. Ticket::Frontend::CustomerTicketZoom###HistoryComment

Description	Value
Description:	Defines the history comment for the ticket zoom action, which gets used for ticket history in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'HistoryComment'} = '';`

2.61.6. Ticket::Frontend::CustomerTicketZoom###Priority

Description	Value
Description:	Allows customers to change the ticket priority in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'Priority'} = '1';`

2.61.7. Ticket::Frontend::CustomerTicketZoom###PriorityDefault

Description	Value
Description:	Defines the default priority of follow up customer tickets in the ticket zoom screen in the customer interface.

Description	Value
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'PriorityDefault'} = '3 normal';`

2.61.8. Ticket::Frontend::CustomerTicketZoom###State

Description	Value
Description:	Allows choosing the next compose state for customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'State'} = '1';`

2.61.9. Ticket::Frontend::CustomerTicketZoom###StateDefault

Description	Value
Description:	Defines the default next state for a ticket after customer follow up in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'StateDefault'} = 'open';`

2.61.10. Ticket::Frontend::CustomerTicketZoom###StateType

Description	Value
Description:	Defines the next possible states for customer tickets in the customer interface.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'StateType'} = [ 'open', 'closed'`

Description	Value
	`];`

2.61.11. Ticket::Frontend::CustomerTicketZoom###AttributesView

Description	Value
Description:	Shows the activated ticket attributes in the customer interface (0 = Disabled and 1 = Enabled).
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	1
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'AttributesView'} = { 'Owner' => '0', 'Priority' => '1', 'Queue' => '1', 'Responsible' => '0', 'SLA' => '0', 'Service' => '0', 'State' => '1', 'Type' => '0' };`

2.61.12. Ticket::Frontend::CustomerTicketZoom###DynamicField

Description	Value
Description:	Dynamic fields shown in the ticket zoom screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled.
Group:	Ticket
SubGroup:	Frontend::Customer::Ticket::ViewZoom
Valid:	1
Required:	0
Config-Setting:	`$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'DynamicField'} = {};`

2.62. Frontend::Queue::Preferences

2.62.1. QueuePreferences###Comment2

Description	Value
Description:	Parameters of the example queue attribute Comment2.
Group:	Ticket
SubGroup:	Frontend::Queue::Preferences
Valid:	0
Required:	0
Config-Setting:	`$Self->{'QueuePreferences'}->{'Comment2'} = { 'Block' => 'TextArea',`

Description	Value

	`'Cols' => '50',`
	`'Desc' => 'Define the queue comment 2.',`
	`'Label' => 'Comment2',`
	`'Module' => 'Kernel::Output::HTML::QueuePreferencesGeneric',`
	`'PrefKey' => 'Comment2',`
	`'Rows' => '5'`
	`};`

2.63. Frontend::SLA::Preferences

2.63.1. SLAPreferences###Comment2

Description	Value
Description:	Parameters of the example SLA attribute Comment2.
Group:	Ticket
SubGroup:	Frontend::SLA::Preferences
Valid:	0
Required:	0

Config-Setting:	`$Self->{'SLAPreferences'}->{'Comment2'} = {`
	`'Block' => 'TextArea',`
	`'Cols' => '50',`
	`'Desc' => 'Define the sla comment 2.',`
	`'Label' => 'Comment2',`
	`'Module' => 'Kernel::Output::HTML::SLAPreferencesGeneric',`
	`'PrefKey' => 'Comment2',`
	`'Rows' => '5'`
	`};`

2.64. Frontend::Service::Preferences

2.64.1. ServicePreferences###Comment2

Description	Value
Description:	Parameters of the example service attribute Comment2.
Group:	Ticket
SubGroup:	Frontend::Service::Preferences
Valid:	0
Required:	0

Config-Setting:	`$Self->{'ServicePreferences'}->{'Comment2'} = {`
	`'Block' => 'TextArea',`
	`'Cols' => '50',`
	`'Desc' => 'Define the service comment 2.',`
	`'Label' => 'Comment2',`
	`'Module' => 'Kernel::Output::HTML::ServicePreferencesGeneric',`
	`'PrefKey' => 'Comment2',`
	`'Rows' => '5'`
	`};`

Anhang C. Danksagungen

Wie die meisten OpenSource-Projekte möchten auch wir vielen Leuten für Ihre Hilfe herzlich danken. Dies ist eine offene Liste in der wir sicher den Einen oder Aderen unabsichtlich vergessen haben, sorry dafür! Meldet Euch einfach, wenn Euer Name hier fehlt!

Folgende Personen haben das Projekt besonders unterstützt bzw. helfen immernoch aktiv mit:

- Robert Kehl, der den Win32-Installer für die 1.x Versionen alleine kreiert hat. Dank Dir, Robert!
- Torsten Werner, der sich aktiv um den Debian Installer kümmert und dafür sorgt, dass OTRS auch bequem unter Debian verfügbar ist. Vielen Dank, Torsten!
- Nils Jeppe (Mirror Hamburg, Germany), Bryan Fullerton (Mirro Toronto, Canada), Eberhard Moenkeberg (Mirror Göttingen, Deutschland), Timo Dreger (Mirror Düsseldorf, Deutschland) und Netmonic (Mirror Wien, Österreich), die unseren ftp-Server spiegeln. Vielen Dank an Euch, durch Eure Hilfe ist immer ein schneller Download von OTRS möglich!
- Anja Schneider, die kräftig bei der Übersetzung dieses Admin Handbuches mitgeholfen und die deutschen Texte korrigiert hat. Vielen Dank, Anja, für deine Geduld und deine Mithilfe!
- Aus den OTRS Mailinglisten schöpfen wir viele Ideen für Weiterentwicklungen, erhalten Patches oder Fixes für Fehler und die Benutzer bekommen sehr guten Support. Vielen Dank an alle auf den Listen für Eure aktive Mithilfe!

Weiterhin möchten wir den folgenden Personen herzlich danken:

- Martin Scherbaum
- Carsten Gross
- Harald Müller
- Stefan Schmidt
- Milisav Radmanic
- Uli Hecht
- Norman Walsh
- Heiko Baumann
- Atif Ghaffar
- Pablo Ruiz Garcia
- Dan Rau
- Christoph Kaulich
- Mark Jackson
- Diane Shieh
- Bernard Choppy

- Carl Bailey
- Phil Davis
- Edwin D. Vinas
- Lars Müller
- Vladimir Gerdjikov
- Fred van Dijk
- Sebastien Guilbaud
- Wiktor Wodecki
- Arnold Ligtvoet
- Antti Kämäräinen
- Nicolas Goralski
- Gilberto Cezar de Almeida
- Jorge Becerra
- Eddie Urenda
- Stella Power
- Andreas Haase
- Reiner Keller
- Covert Jake
- Moshe Leibovitch
- Bjärn Jacke
- Remo Catelotti
- Alfons Obermeyer
- Michael Rech
- Danie Theron
- Richard Paradies
- Art Powell

Anhang D. GNU Free Documentation License

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document, that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters), and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed

of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats, suitable for input to text formatters. A copy made in an otherwise Transparent file format, whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include: plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include: PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location, containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously, at no charge, using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location, until at least one year

after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version, as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License, for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. Übersetzung

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections, in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

. How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation

License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.