



OTRS
Open Technology
Real Services

Documentation

OTRS 6 - Admin Manual

Build Date:

2016-03-14

OTRS 6 - Admin Manual

Copyright © 2003-2016 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, Rolf Schmidt, Burchard Steinbild, Michael Thiessmeier, 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.

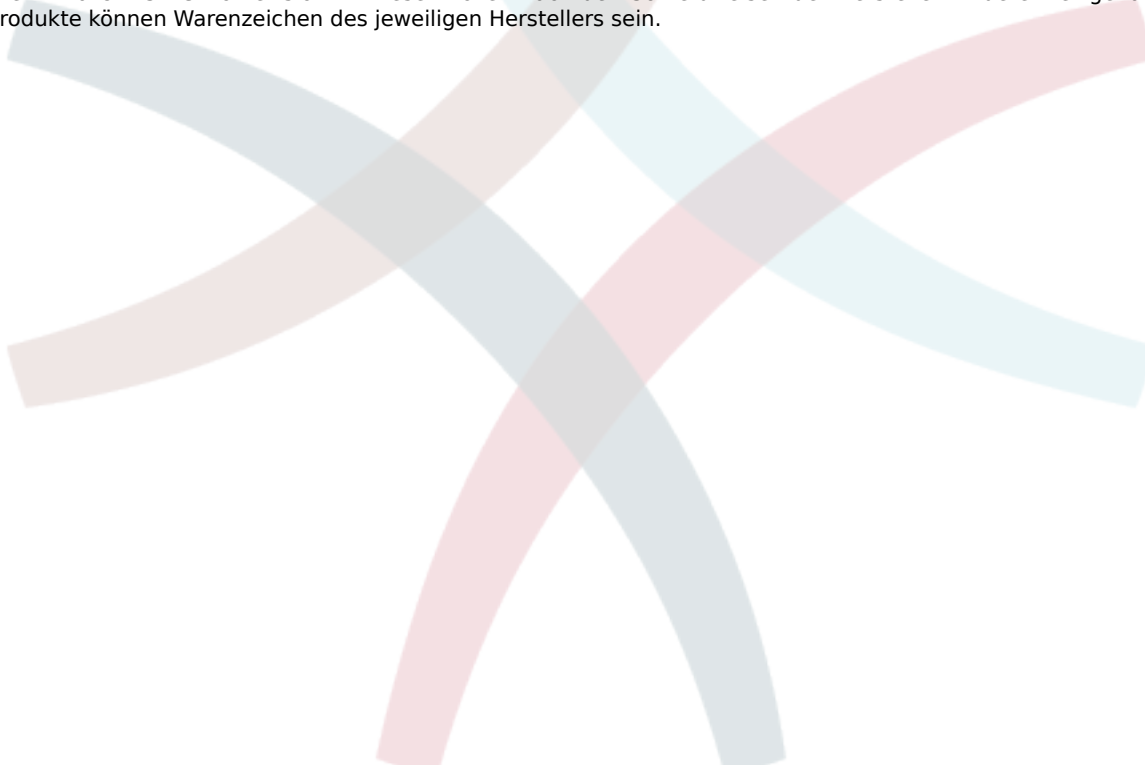
Der Quellcode dieses Dokuments kann bei [github](#) gefunden werden, im Repository [doc-admin](#). Beiträge sind mehr als willkommen. Sie können auch bei der Übersetzung in Ihre Sprache helfen, und zwar bei [Transifex](#).

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	xi
1. Einführung	1
1. Trouble-Ticket-Systeme - die Grundlagen	1
1.1. Was ist ein Trouble-Ticket-System, und warum benötigen Sie eins?	1
1.2. Was ist ein Trouble-Ticket?	2
2. OTRS-Helpdesk	2
2.1. Grundlagen	2
2.2. Features	3
2.3. Hardware and Software Requirements	9
2.4. Community	10
2.5. Kommerzieller Support und Dienstleistungen für OTRS	10
2. Installation	11
1. The Simple Way - Installation of Pre-Built Packages	11
1.1. Installation des RPM auf einem SuSE-Linux-Server	11
1.2. Installation von OTRS auf einem REd Hat Enterprise Linux oder CentOS-System	13
1.3. Installation von OTRS auf einem Debian- oder Ubuntu-System	17
2. Installation From Source (Linux, Unix)	17
3. Using the Web Installer	22
4. OTRS auf Windows	27
4.1. Migration existierender Windows-Installationen nach Linux	27
5. Upgrading OTRS from 4 to 5	38
6. Additional Applications	45
6.1. FAQ	45
3. First Steps	46
1. Agent Web Interface	46
2. Customer Web Interface	46
3. Public Web Interface	47
4. First Login	47
5. The Web Interface - an Overview	48
6. Die Übersichtsseite	50
7. What is a Queue?	56
8. What is the Queue Overview?	57
9. User Preferences	58
4. Administration	60
1. The Administration Area of OTRS	60
1.1. Grundlagen	60
1.2. Benutzer, Gruppen und Rollen	60
1.3. Kundenbenutzer und Kundengruppen	67
1.4. Queues	70
1.5. Salutations, Signatures, Attachments and Templates	71
1.6. Automatische Antworten	77
1.7. System Email Addresses	79
1.8. Ticket-Benachrichtigungen	79
1.9. S/MIME	82
1.10. PGP	83
1.11. Status	83
1.12. Die SysConfig (Verwaltung der Systemkonfiguration)	84
1.13. Using Mail Accounts	85
1.14. Filtering Incoming Email Messages	85
1.15. Executing Automated Jobs with the GenericAgent	88
1.16. Administrative Messages	89
1.17. Sitzungsverwaltung	90
1.18. Systemwartung	91
1.19. System Log	92

1.20. SQL Queries via the SQL Box	93
1.21. Paketverwaltung	93
1.22. Web-Services	94
1.23. Dynamische Felder	94
2. Systemkonfiguration	95
2.1. Die Konfigurationsdateien von OTRS	95
2.2. Configuring the System Through the Web Interface	95
3. Backing Up the System	96
3.1. Datensicherung	97
3.2. Wiederherstellung der Daten	97
4. E-Mail-Einstellungen	98
4.1. Sending/Receiving Emails	98
4.2. Secure Email with PGP	103
4.3. Secure Email with S/MIME	106
5. Using External backends	109
5.1. Customer Data	109
5.2. Customer User Backend	109
5.3. Backends to Authenticate Agents and Customers	116
5.4. Customizing the Customer Self-Registration	121
6. Ticket-Einstellungen	123
6.1. Ticket Status	123
6.2. Ticket Priorities	126
6.3. Ticket-Verantwortlicher und Beobachten von Tickets	127
7. Date and Time Related Functions	129
7.1. Arbeitszeiten, Feiertage und Zeitzonen für das System festlegen	129
7.2. Automatische Ticket-"Freigabe" (unlock)	130
8. Customizing the PDF Output	131
9. Statistiken	131
9.1. Statistics Configuration and Usage	131
9.2. Statistics System Administration	137
10. Dynamische Felder	138
10.1. Einführung	138
10.2. Konfiguration	138
11. Generic Interface	155
11.1. Generische Schnittstellenebenen	155
11.2. Generic Interface Kommunikationsfluss	157
11.3. Web-Services	160
11.4. Web Service - Grafische Oberfläche	160
11.5. Webdienst-Befehlszeilenoberfläche	179
11.6. Webservice Konfiguration	179
11.7. Verbinder	185
12. The OTRS Daemon	201
12.1. OTRS Daemon Graphical Interface	203
12.2. OTRS Daemon Command Line Interface	203
5. Anpassung	205
1. Access Control Lists (ACLs)	205
1.1. Einführung	205
1.2. Definition	205
1.3. Beispiele	206
1.4. Referenz	212
2. Prozess-Management	215
2.1. Einführung	215
2.2. Beispiel Prozess	215
2.3. Implementierung des Beispiels	215
2.4. Prozesskonfigurationsempfehlung	241
3. Localization of the OTRS Front End	262
6. Leistungsverbesserung	263
1. OTRS	263

1.1. TicketIndexModule	263
1.2. TicketStorageModule	263
1.3. Tickets archivieren	264
1.4. Cache	265
2. Datenbank	265
2.1. MySQL	265
2.2. PostgreSQL	265
3. Webserver	266
3.1. Persistente Datenbankverbindungen	266
3.2. Vorgeladene Module - startup.pl	266
3.3. Perl Module bei Änderung neu laden	266
3.4. Die richtige Strategie wählen	266
3.5. mod_gzip/mod_deflate	266
A. Weitere Quellen	267
B. Referenz der Konfigurationseinstellungen	268
1. CloudService	268
2. Daemon	269
3. DynamicFields	279
4. Framework	283
5. GenericInterface	392
6. ProcessManagement	401
7. Ticket	409
C. GNU Free Documentation License	591
0. PREAMBLE	591
1. APPLICABILITY AND DEFINITIONS	591
2. VERBATIM COPYING	592
3. COPYING IN QUANTITY	592
4. MODIFICATIONS	593
5. COMBINING DOCUMENTS	594
6. COLLECTIONS OF DOCUMENTS	594
7. AGGREGATION WITH INDEPENDENT WORKS	595
8. Übersetzung	595
9. TERMINATION	595
10. FUTURE REVISIONS OF THIS LICENSE	595
How to use this License for your documents	596

Abbildungsverzeichnis

2.1. Welcome screen	23
2.2. GNU Affero General Public License	23
2.3. Datenbank-Auswahl	24
2.4. Database credentials	24
2.5. Database settings	25
2.6. Successful database setup	25
2.7. System settings	26
2.8. Mail configuration	26
2.9. Web installer final screen	27
2.10. Download des OTRSCloneDB-Skript - Screenshot	28
2.11. Installieren des OTRSCloneDB-Skripts - Screenshot	29
2.12. Zieldatenbankpasswort holen - Bildschirmfoto	31
2.13. Konfiguration der OTRSCloneDB Sysconfig 1 - screenshot	32
2.14. Konfiguration der OTRSCloneDB Sysconfig 2 - screenshot	33
2.15. Ausführen des OTRSCloneDB-Skript 1 - screenshot	34
2.16. Run OTRSCloneDB script 2 - screenshot	35
2.17. Run OTRSCloneDB script 3 - screenshot	36
3.1. Login screen of the agent interface	46
3.2. Login screen of the customer interface	47
3.3. Öffentliche Weboberfläche	47
3.4. Neues Passwort beantragen	48
3.5. Dashboard of the agent interface	49
3.6. Fußzeile	50
3.7. Übersichtsseiten-Widgets	51
3.8. Ticket Ereigniskalender Widget	53
3.9. Dashboard Settings	55
3.10. Queue-Ansicht (Standard) für Agenten	57
3.11. Visuelle Alarme der Queueansicht	57
3.12. Agent's personal preferences	58
3.13. Customer's personal preferences	59
4.1. OTRS Administration Overview Screen	60
4.2. Agenten-Verwaltung	61
4.3. Adding a new agent	61
4.4. Gruppen-Verwaltung	62
4.5. Agent <-> group management	62
4.6. Change the groups an agent belongs to	63
4.7. Change the agents that belong to a specific group	63
4.8. Rollenverwaltung	65
4.9. Adding a new role	66
4.10. Change the roles associated with an agent	66
4.11. Change the agents associated with a specific role	66
4.12. Manage roles-groups relations	67
4.13. Change group relations for a role	67
4.14. Change role relations for a group	67
4.15. Kundenverwaltung	68
4.16. Adding a customer	68
4.17. Customer-Group relations management	69
4.18. Gruppenzuordnungen eines Kunden bearbeiten	69
4.19. Kundenzuordnungen einer Gruppe bearbeiten	69
4.20. Queue-Verwaltung	70
4.21. Adding a new queue	70
4.22. Verwaltung der Anreden	71
4.23. Adding a new salutation	72
4.24. Signatures management	72
4.25. Adding a new signature	73

4.26. Attachments management	73
4.27. Adding a new attachment	74
4.28. Anhänge zu Vorlagen zuordnen.	74
4.29. Anlagenzuordnungen einer Vorlage verändern	74
4.30. Vorlagen-Zuordnungen für einen Anhang verändern	74
4.31. Template management	75
4.32. Adding a template	76
4.33. Template-Queue relations management	76
4.34. Ändern der Queue Zuordnung für eine Vorlage	76
4.35. Antwort-Zuordnungen einer Queue ändern	76
4.36. Auto response management	77
4.37. Adding an auto response	77
4.38. Queue <-> auto response relations management	78
4.39. Change auto response relations for a queue	79
4.40. System email addresses management	79
4.41. Adding a system email address	79
4.42. Ticket notification management	80
4.43. Customizing a notification	81
4.44. Customizing a notification's recipients	81
4.45. Customizing notification methods	82
4.46. S/MIME management	83
4.47. PGP management	83
4.48. State management	83
4.49. Das grafische Konfigurationsfrontend von OTRS (SysConfig)	85
4.50. Verwaltung von Mail-Konten	85
4.51. PostMaster filter management	86
4.52. Add a PostMaster filter	88
4.53. Jobliste des GenericAgent	88
4.54. Anlegen eines GenericAgent-Jobs	89
4.55. Admin notification screen	90
4.56. Sitzungsverwaltung	90
4.57. Sitzungsdetails	91
4.58. The system maintenance overview screen with some scheduled periods	91
4.59. The system maintenance edit screen	92
4.60. System Log	92
4.61. SQL Box	93
4.62. Paketverwaltung	93
4.63. Die grafische Konfigurationsoberfläche für Webservices	94
4.64. The dynamic fields overview screen with some dynamic fields	94
4.65. Die grafische System-Konfigurationsoberfläche	96
4.66. Einen E-Mail-Account hinzufügen	98
4.67. Changing the Responsibility of a ticket in its zoomed view	127
4.68. Pop-up dialog to change a ticket's responsibility	128
4.69. Subscribing to watching a ticket in its zoomed view	128
4.70. Unsubscribing from watching a ticket in its zoomed view	129
4.71. Watched tickets view	129
4.72. Overview of the standard statistics.	132
4.73. Viewing a specific statistic.	132
4.74. Adding a new statistic, first step.	133
4.75. Adding a new statistic, second step.	134
4.76. Configuring the x-axis of a statistic.	135
4.77. Configuring the y-axis of a statistic.	135
4.78. Configuring the data filter of a statistic.	135
4.79. Configuring the data filter of a statistic.	136
4.80. Statistics import	136
4.81. Leere Übersichtsseite der dynamischen Felder	139
4.82. Konfigurationsdialog für Textfelder in dynamischen Feldern	141
4.83. Konfigurationsdialog für Textfelder in dynamischen Feldern	142

4.84. Konfigurationsdialog für Checkboxen in dynamischen Feldern	142
4.85. Konfigurationsdialog für Einfachauswahl in dynamischen Feldern	144
4.86. Konfigurationsdialog für Mehrfachauswahlen in dynamischen Feldern	145
4.87. Konfigurationsdialog für Datum in dynamischen Feldern.	146
4.88. Konfigurationsdialog für Datum / Zeit in dynamischen Feldern	147
4.89. Übersichtsseite der dynamischen Felder mit Beispieldaten	148
4.90. Field1 in New Phone Ticket Screen	149
4.91. Field1 in New Phone Ticket Screen as mandatory	150
4.92. Several fields in New Phone Ticket Screen as mandatory	151
4.93. Some deactivated fields in New Phone Ticket Screen as mandatory	152
4.94. Field1 in Ticket Zoom Screen	152
4.95. Field1 in Ticket Overview Small Screen	153
4.96. Field1 in User preferences screen	155
4.97. The graphical interface layers	156
4.98. Web services overview	161
4.99. Web services add	162
4.100. Webservice kopieren	163
4.101. Web services export	164
4.102. Web services import	165
4.103. Web service history	166
4.104. Webservice löschen	167
4.105. Web service debugger	168
4.106. Web services change	169
4.107. Web service provider network transport (HTTP::SOAP)	170
4.108. Web service provider network transport (HTTP::REST)	172
4.109. Web service operation	173
4.110. Web service requester network transport (HTTP::SOAP)	174
4.111. Web service provider network transport (HTTP::REST)	176
4.112. Web service invoker	177
4.113. Web service mapping	178
4.114. Daemon notification	203
4.115. Start Daemon	203
5.1. ACL 100-Example-ACL	207
5.2. ACL 102-Example-ACL	208
5.3. ACL 102-Second-Example-ACL	209
5.4. ACL 103-Third-ACL-Example	210
5.5. ACL 104-Only-Hardware-Services-for-HW-Queues-ACL	210
5.6. ACL 105-Disallow-Process-For-CustomerID	211
5.7. OTRS Admin screen - System Administration	219
5.8. Schaltfläche: Neuen Prozess erstellen	219
5.9. Neuen Prozess hinzufügen	220
5.10. Neue Aktivitätsdialogschaltfläche erstellen	220
5.11. Neuen Aktivitätsdialog hinzufügen	221
5.12. Felddetails bearbeiten (Artikel)	221
5.13. Neuen Übergang hinzufügen	224
5.14. Neuen Übergang hinzufügen	224
5.15. Erstellen Sie eine neue Schaltfläche "Übergangs-Aktion"	225
5.16. Neue Übergangs-Aktion hinzufügen	226
5.17. Neue Aktivitätsschaltfläche erstellen	227
5.18. Ziehen Sie die erste Aktivität in die Arbeitsfläche	229
5.19. Ziehen Sie die zweite Aktivität in die Arbeitsfläche.	229
5.20. Ziehen Sie einen Übergang in die Arbeitsfläche	230
5.21. Verbinden Sie die Aktivitäten unter Verwendung von Übergängen	230
5.22. Zuweisen von Übergangs-Aktionen	231
5.23. vollständiger Buchbestellprozesspfad	233
A.1. Fehlerverfolgungssystem	267

Tabellenverzeichnis

4.1. Standardmäßig vorhandene Gruppen in OTRS	61
4.2. Rights associated with OTRS groups	63
4.3. Weitere Berechtigungsgruppen	64
4.4. Events for auto responses	78
4.5. Funktion der verschiedenen X-OTRS-Header	86
4.6. Die folgenden Felder werden zu dem System hinzugefügt:	140
A.1. Mailinglisten	267



Liste der Beispiele

4.1. Aussortierung von Spammails in eine bestimmte Queue	88
4.2. Routing via Procmail Using otrs.Console.pl	99
4.3. .fetchmailrc	100
4.4. Example jobs for the filter module	
Kernel::System::PostMaster::Filter::Match	101
4.5. Beispiel-Job für das Filtermodul Kernel::System::PostMaster::Filter::CMD	101
4.6. Example job for the filter module	
Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition	102
4.7. Konfiguration eines Datenbank-Kunden-Backends	110
4.8. Using Company Tickets with a DB Backend	112
4.9. Konfiguration eines LDAP Kunden-Backends	112
4.10. Verwendung von Firmentickets mit einem LDAP-Backend	114
4.11. Gleichzeitige Einbindung mehrerer verschiedener Kunden Backends	114
4.12. Agentenauthentifizierung gegen ein Datenbank-Backend	117
4.13. Agentenauthentifizierung gegen ein LDAP Backend	117
4.14. Agentenauthentifizierung über HTTPBasic	119
4.15. Agentenauthentifizierung gegen ein Radius-Backend	119
4.16. Kundenauthentifizierung gegen ein Datenbank-Backend	119
4.17. Kundenauthentifizierung gegen ein LDAP Backend	120
4.18. Kundenauthentifizierung über HTTPBasic	120
4.19. Kundenauthentifizierung gegen ein Radius-Backend	121
4.20. Default statistics permission group	134
4.21. Customized statistics permission group	134
4.22. Feld1 im Dialog NewPhoneTicket anzeigen	149
4.23. Activate Field1 in New Phone Ticket Screen as mandatory. Feld1 als verpflichtend in Dialog NewPhoneTicket setzen.	149
4.24. Aktivieren von verschiedenen Feldern in der Oberfläche 'Neues Telefonticket'	150
4.25. Deaktivieren von Feldern in der Oberfläche 'Neues Telefonticket'	151
4.26. Aktiviert Feld1 in der TicketZoom-Oberfläche.	152
4.27. Aktiviert Feld1 in der kleinen Ticketübersicht.	153
4.28. Aktiviert Feld1 im Event TicketCreate.	153
4.29. Aktiviert Field1 in den Benutzereinstellungen	154
4.30. Example to start the OTRS Daemon	203
4.31. Example to list all configured daemons	204
4.32. Example to a summary of all daemon tasks	204
5.1. ACL, die das Verschieben von Tickets in eine Queue nur bei einer Ticket-Priorität von 5 erlaubt.	206
5.2. ACL, die das Verschieben von Tickets in eine Queue nur bei einer Ticket-Priorität von 5 erlaubt.	207
5.3. ACL, die das Schließen von Tickets in der raw-Queue verbietet und den Schließen-Schalter ausblendet.	208
5.4. ACL entfernt immer den Status Erfolgreich geschlossen	209
5.5. ACL, die nur Hardware-Services für Tickets anzeigt, die in Queues erstellt wurden, die mit "HW" beginnen.	210
5.6. ACL to restrict a Process in the customer frontend using the CustomerID.	211
5.7. Die Liste zeigt alle möglichen und wichtigen ACL-Einstellungen	212

Vorwort

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

Die folgenden Kapitel beschreiben 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.

This book continues to be a work in progress, given a moving target on new releases. We need your feedback in order to make this a high quality reference document: one that is usable, accurate, and complete. Please write to us if you find content missing in this book, if things are not explained sufficiently, or even if you see spelling mistakes, grammatical errors, or typos. Any kind of feedback is highly appreciated and should be made via our bug tracking system on <http://bugs.otrs.org>. Thanks in advance for your contributions!



Kapitel 1. Einführung

1. Trouble-Ticket-Systeme - die Grundlagen

Dieses Kapitel enthält eine Kurzeinführung in Ticketsysteme, zusammen mit einer Erklärung des Kernkonzepts eines "Trouble-Tickets". Ein kurzes Beispiel illustriert die Vorteile der Nutzung eines solchen Systems.

1.1. 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 einsparen können.

Max Mustermann ist ein Hersteller von Video-Rekordern. Herr Mustermann empfängt viele E-Mails von Kunden, die Hilfe benötigen. An manchen Tagen ist er aber nicht in der Lage, zeitnah auf Nachrichten zu reagieren. Einige Kunden werden dann ungeduldig und schreiben eine zweite E-Mail mit derselben Frage. Alle diese Support-Anfragen werden unsortiert im Posteingang von Herrn Mustermanns E-Mail-Programm gespeichert, womit er die Anfragen auch abarbeitet.

Weil Herr Mustermann nicht schnell genug auf all diese E-Mails antworten kann, wird er dabei von den Entwicklern Meier und Schulze unterstützt, die das gleiche E-Mail-System nutzen und auf den gleichen Posteingang zugreifen. Sie berücksichtigen dabei aber nicht, dass Herr Mustermann oft zwei identische Anfragen vom gleichen Kunden erhält. Manchmal kommt es vor, dass beide unabhängig voneinander auf die gleiche Anfrage antworten, was zur Folge hat, dass der Kunde zwei unterschiedliche Antworten erhält. Des Weiteren weiß Herr Mustermann wiederum nicht, was seine Kollegen dem Kunden genau geantwortet haben. Außerdem kennt er weder die Details der Anfragen oder die Lösungsvorschläge seiner Kollegen, oder wie oft bestimmte Probleme bei verschiedenen Kunden auftreten und wie viel Zeit und Geld er für die Unterstützung von Kunden bereits verbraucht hat.

In einem Meeting erfährt Herr Mustermann von sogenannten Trouble-Ticket-Systemen und wie sie Herrn Mustermanns Probleme mit den Kunden-Anfragen lösen können. Nachdem er sich im Internet erkundigt hat, entscheidet sich Herr Mustermann, das frei verfügbare Ticket-System "OTRS" auf einem Rechner zu installieren, der sowohl für seine Kunden, als auch für ihn und seine Mitarbeiter über das Internet erreichbar ist. Jetzt landen die Kunden-Anfragen nicht mehr in seinem persönlichen Posteingang, sondern direkt in einem Mail-Account, der für OTRS eingerichtet wurde. Das Ticket-System speichert alle eingehenden E-Mails in seiner Datenbank. Für jede Anfrage generiert das System automatisch eine Antwort, so dass der Kunde weiß, dass seine Anfrage angekommen ist und bald beantwortet werden wird. OTRS erstellt dabei für jede Anfrage eine eindeutige Referenz - die Ticket-Nummer. Die Kunden sind jetzt zufriedener, da sie wissen, dass ihre Anfragen berücksichtigt werden und senden keine Mehrfach-Anfragen mehr. Herr Mustermann und seine Kollegen können sich jetzt mit einem einfachen Web-Browser in OTRS anmelden und die Anfragen bearbeiten. Weil das System Tickets sperrt, die gerade beantwortet werden, wird keine Nachricht mehr versehentlich doppelt bearbeitet.

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 handelt es sich dabei um ein verkürztes Beispiel-Szenario, das nur einige der Möglichkeiten und Features von Trouble Ticket-Systemen aufzeigt. Aber wenn Ihr Unternehmen eine große Anzahl von Kunden-Anfragen bewältigen muss und verschiedene Service-Mitarbeiter zu verschiedenen Zeiten auf diese Anfragen reagieren können müssen, kann ein Ticket-System sie dabei optimal unterstützen. Es hilft dabei, Arbeitsabläufe und -Prozesse zu vereinheitlichen, effizienter zu machen und Ihre Produktivität insgesamt zu steigern. Ein Ticket-System hilft Ihnen, Ihren Support- oder Help-Desk-Bereich flexibel zu strukturieren. Kommunikations-Abläufe zwischen Kunden und Service-Mitarbeitern werden transparenter. Das Resultat ist eine gesteigerte Service-Effizienz. Und zweifelsohne bedeuten zufriedene Kunden auch finanziell bessere Ergebnisse für Ihr Unternehmen.

1.2. 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 eine Mehrfachdosierung von Medikamenten. Ist die Krankheit besiegt und der Patient entlassen, wird das Krankenblatt archiviert.

Im OTRS werden Trouble-Tickets, also die Krankenblätter aus dem obigen Beispiel, als normale E-Mails behandelt und gespeichert. Schickt ein Kunde eine Anfrage an das Trouble-Ticket-System, wird das Krankenblatt eingerichtet - ein neues Ticket wird geöffnet. Die Antwort eines Mitarbeiters auf die Anfrage kann als Eintrag eines Arztes gesehen werden, eine erneute Antwort (bzw. Anfrage des Kunden auf dasselbe 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, wird es erneut geöffnet und um 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 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.

2. OTRS-Helpdesk

In diesem Abschnitt werden die Features des Open Ticket Request Systems (OTRS) vorgestellt. Des Weiteren wird näher auf die Systemanforderungen von OTRS eingegangen und erläutert, wie Kontakt zur OTRS-Community aufgenommen werden kann bzw. wie kommerzieller Support erhältlich ist.

2.1. Grundlagen

OTRS Help Desk (OTRS) ist eine Webanwendung, die auf einem Webserver installiert und mit einem Browser genutzt wird.

OTRS ist in mehrere Komponenten aufgeteilt. Die Basis bildet ein Framework, das alle grundlegenden Komponenten für die Applikation und das Trouble-Ticket-System enthält. Es können zusätzliche Komponenten wie das OTRS::ITSM-Modul, eine Knowledge-Base (FAQ) oder Netzwerk-Überwachungs-Lösungen installiert werden.

2.2. Features

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

2.2.1. User Interface

- OTRS enthält getrennte, moderne Benutzeroberflächen für Agenten und Kunden.
- Es kann mit jedem modernen Web-Browser und auf mobile Plattformen genutzt werden und ist für Retina-Bildschirme optimiert.
- Die Benutzeroberfläche kann mit Skins den eigenen Vorstellungen angepasst werden (z.B. gemäß der Corporate Identity Ihres Unternehmens).
- Umfangreiches und anpassbare Agenten-Übersichtsseite mit persönlichen Ticket-Übersichten und Unterstützung für grafische Statistiken.
- Eine erweiterbare Reporting-Engine stellt verschiedene Statistiken und zeitlich planbare Reports zur Verfügung.
- Mit dem Prozess-Management ist es möglich, eigene Ticket-basierte Masken und Prozesse zu definieren (Ticket-Workflows).
- OTRS hat eine eingebaute Rechte-Verwaltung, die um hochgranulare Zugriffskontrolllisten (ACLs) ergänzt werden kann.
- Unterstützung in mehr als 30 Sprachen und verschiedenen Zeit Zonen.

2.2.2. Email Interface

- Unterstützung für MIME-E-Mails mit Anhängen.
- Automatische Umwandlung von HTML- in reine Text-Nachrichten (höhere Sicherheit vor schädlichen Inhalten und schneller durchsuchbar).
- Eingehende E-Mails können gefiltert und mithilfe komplexer Regeln vorverarbeitet werden, zum Beispiel zum Erkennen von Spam-Nachrichten und zur Verteilung in passende Queues.
- Unterstützung für PGP- und S/MIME-Standards mit Schlüssel- und Zertifikatsverwaltung und E-Mail-Verarbeitung.
- Automatische Antworten, für jede Queue konfigurierbar.
- E-Mail-Benachrichtigungen für Agenten über neue Tickets, Follow-ups oder freigegebene Tickets.
- Es ist möglich, eigene Definitionen zur Erkennung von Folge-Kommunikation zu treffen (z.B. Anruf#, Ticket# oder Anfrage#). Es stehen verschiedene Nummern-Generatoren (Datumsbasiert, zufällig, etc.) zur Verfügung, außerdem können eigene Generatoren verwendet werden. Folge-Kommunikation kann außerdem anhand von In-Reference-To-Header oder externen Ticket-Nummern erkannt werden.

2.2.3. Ticket

- OTRS nutzt Tickets, um interne und externe Kommunikation zu bündeln. Dieses Tickets sind wiederum in Queues organisiert.
- Es gibt verschiedene Möglichkeiten (z.B. Queue-, Status- oder Eskalationsbasiert) und verschiedene Detailgrade (klein/mittel/Vorschau) zum Betrachten von Tickets im System.
- Die Ticket Historie speichert alle Änderungen am Ticket.
- Tickets können zum Beispiel durch Antworten, Weiterleiten, Umleiten, Verschieben in eine andere Queue, Ändern von Attributen (Status, Priorität, etc.), Sperren oder Einbuchungen von Zeiteinheiten bearbeitet werden. Es ist auch möglich, mehrere Tickets gleichzeitig zu bearbeiten (Stapelverarbeitung).
- "Warten bis"-Zeiten, Eskalationszeiten und das SLA-Management ermöglichen zeitgesteuerte Planung und Einschränkungen für Tickets.
- Tickets können mit anderen Tickets oder generischen Objekten (z.B. FAQ-Artikeln) verlinkt werden.
- Automatische und zeitgesteuerte Aktionen können mit Hilfe eines sog. "GenericAgent" auf Tickets ausgeführt werden.
- OTRS enthält eine mächtige Suchfunktion, die komplexe und Volltextsuchen auf Tickets ermöglicht.

2.2.4. System

- OTRS läuft auf vielen Betriebssystemen (Linux, Solaris, AIX, FreeBSD, OpenBSD, Mac OS 10.x) und unterstützt etliche Datenbanksysteme für das zentrale OTRS-Backend (MySQL, PostgreSQL, Oracle, MSSQL).
- Der OTRS-Kern kann durch Installation von Paketen erweitert werden. Viele Pakete sind frei verfügbar (wie FAQ, OTRS::ITSM etc.), zusätzlich stehen weitere Pakete als Feature-AddOns für Service-Kunden der OTRS-Gruppe zur Verfügung.
- Integration von externen Backends für Kundendaten, z.B. Active Directory, eDirectory oder OpenLDAP. Kunden können sich per Datenbank, LDAP, HTTPAuth oder Radius authentifizieren.
- Mit dem GenericInterface ist es leicht möglich, OTRS mit anderen Web-Services zu verbinden. Einfache Web-Services können ohne Programmieraufwand angebunden werden, komplexe Szenarios mithilfe von eigenen Anpassungen. Der OTRS Ticket-Connector erlaubt die Erstellung und Aktualisierung von, sowie die Suche nach Tickets aus Dritt-Applikationen via Web-Service.

Nachfolgend finden Sie einen Überblick über Veränderungen in den letzten Versionen von OTRS.

2.2.5. Neue Features von OTRS 5

2.2.5.1. Produktivität

- OTRS ist nun optimiert für die Nutzung auf Mobilgeräten unterschiedlicher Art und Größe.
- Einfach- und Mehrfachauswahlfelder wurden modernisiert und bieten erweiterte Such- und Filtermöglichkeiten (Danke an Dusan Vuckovic bei Mühlbauer).

- Dateien können nun von überall außerhalb der Anwendung mit Copy&Paste und Drag&Drop zum WYSIWYG-Editor hinzugefügt/hochgeladen werden (in allen Browsern, ohne zusätzliche Erweiterung).
- Verbessertes Benachrichtigungssystem. Es ist nun möglich Ticketbenachrichtigungen mit eigenen Auslösebedingungen und Empfängern zu konfigurieren. Mit OTRS Business Solution™ können Benachrichtigungen auch per SMS oder Notification Web View versendet werden. Letzteres ist eine spezielle Maske in OTRS, in der alle Benachrichtigungen des Agenten angezeigt werden. Damit kann OTRS völlig ohne E-Mail-Programm verwendet werden.
- Statistics received a new graphical user interface which is much better accessible and helps to create great statistics quickly and easily.
- Additionally, statistics support the new time periods “quarter” and “half-year”
- It is now possible to group action menu items in the ticket zoom screen. Less often used items can be grouped in a submenu, improving screen usage and clarity.
- Ticket overviews can now display customer company data, thanks to Renée Bäcker.
- The ticket process TransitionAction “TicketCreate” can now create tickets without articles.

2.2.5.2. Skalierbarkeit & Performance

- The new OTRS Daemon handles all asynchronous and periodic tasks and replaces all previous OTRS cron jobs. In a clustered environment the load is automatically distributed over the nodes.
- It is now possible to specify multiple readonly mirror (slave) databases for expensive computations such as statistics or fulltext searches to distribute the load among these database servers.

2.2.5.3. Security

- A new two-factor authentication layer allows added login security.
 - If entering a fixed username and password doesn't satisfy your requirements, you can now additionally use the open standard for time based one-time passwords ([RFC 6238](#), also known as Google Authenticator).
 - After having enabled the two-factor authentication, agents and customers can add a shared secret to their preferences and immediately start logging in using one-time passwords created by a compatible method of their choice (e.g. the Android Google Authenticator app).

2.2.5.4. Working with External Systems

- A new XSLT based GenericInterface mapping module allows for arbitrarily complex user-defined data mapping.

2.2.5.5. Installation & Administration

- The new OTRS console makes working on the commandline easy and fun. All commands have a consistent interface, useful documentation and provide helpful colored output.
- Administrators can now specify a minimum log level to reduce logging volume, thanks to Renée Bäcker.

- Overview screens in the admin area now show invalid entities in gray, making it easy to focus on active elements.

2.2.6. New Features of OTRS 4

2.2.6.1. Produktivität

- Ein aufgeräumteres flacheres Design wurde implementiert.
- Agenten können nun direkt auf eine Notiz antworten. Die Original-Notiz wird in der neuen Notiz zitiert.
- Agenten können nun in allen Oberflächen, mit internen Notizen, Vorlagen verwenden.
- In Ticket-Aktions-Oberflächen (wie Notiz, Besitzer etc.) können nun Aktionen ausgeführt werden, ohne das ein Artikel am Ticket erstellt wird (konfigurierbar).
- Neue Ticketübersicht basierend auf "MyServices" die ein Agent abonieren kann. Benachrichtigungsoptionen für neue Tickets und FollowUps basierend auf "Meine Queues", "MyServices" oder einer Kombination aus beiden.
- OTRS kann nun Tickets mit tausenden Artikeln darstellen.
- Wenn auf einen Kunden in der "Online-Kundenliste" in der Übersichtsseite geklickt wird, wird man direkt zum Kundeninformationszentrum weitergeleitet.
- Agenten können nun ihr Hauptmenü mittels Drag&Drop neu ordnen.
- Agenten und Kunden können Tickets nun nach Anhängen durchsuchen.
- Neues Übersichtsseiten-Widget für laufende Prozesstickets.
- Neue Suchoption um nach der letzten Änderungszeit eines Tickets zu suchen.
- Eine neue Oberfläche für ausgehende E-Mails auf ein Ticket, das keine Antworten enthält, wurde hinzugefügt.

2.2.6.2. Skalierbarkeit & Performance

- OTRS 4 kann mehr gleichzeitige Benutzer/Anfrage auf der gleichen Hardware verwalten und die Antwortzeiten für einzelne Anfragen sind ebenfalls schneller, besonders für Seiten mit vielen Daten.

2.2.6.3. Arbeiten mit externen Systemen

- Das Generic Interface unterstützt nun auch HTTP REST als Netzwerkübertragungsprotokoll.

2.2.6.4. Installation & Administration

- Postmaster-Filter sind nicht länger auf 4 Treffer-/Setz-Felder beschränkt. Sie können jetzt eine frei konfigurierbare Menge von Feldern besitzen (Standard 12, bis zu 99).
- Eine neue Konfigurationsoption Ticket::MergeDynamicFields macht es möglich, anzugeben welches dynamische Feld zusammengefasst werden soll, wenn ein Ticket mit einem anderen Ticket zusammengefasst wird.
- Neue Optionen hinzugefügt, um dynamische Felder vom Typ Text in Bezug auf Fehlermeldungen (übersetzt) zu überprüfen, wenn sie nicht übereinstimmen.
- Neue Optionen hinzugefügt, um dynamische Felder vom Typ Datum / Datum/Zeit auf zukünftige oder in der Vergangenheit liegende Termine zu beschränken

- OTRS kann so konfiguriert werden, dass ein Ticket automatisch entsperrt wird wenn diesem ein Artikel hinzugefügt wird und der Ticketbesitzer die Abwesenheitsoption aktiviert hat.
- Verknüpfte Tickets mit einem bestimmten Status (bspw. zusammengefasst oder entfernt) können nun über eine Einstellung in der SysConfig ausgeblendet werden.
- Das ACL-Handling wurde verbessert, es ist nun logischer aufgebaut und einfacher zu debuggen.
 - Es wurden neue ACL-Option PossibleAdd hinzugefügt, um Elemente einer möglichen Liste hinzuzufügen, ohne Werte zurücksetzen (wie Possible) .
 - Es wurden für alle ACL-Typen neue ACL-Werte [Not], [NotRegExp], [Notregexp] hinzugefügt.
- Das Prozessmanagement wurde verbessert, es ist nun logischer aufgebaut und einfacher zu debuggen.
 - A new GUID-based entity naming scheme for the OTRS Process configuration makes it possible to safely transfer processes from one system to another without duplicating the entities.
 - Neue Übergangs-Aktion hinzugefügt um ein neues Ticket zu erstellen.
 - Added possibility to define variable Transition Action attributes based on current process ticket values.
- Das Planen eines Systemwartungs-Zeitintervalls ist über die Systemverwaltung im Admin-Interface möglich.
 - Eine Benachrichtigung über eine kommende Systemwartung wird mit einer (konfigurierbaren) Zeitspanne im Voraus angezeigt werden.
 - Wenn eine Systemwartung aktiv ist, wird eine Benachrichtigung in der Agenten- und Kundenoberfläche angezeigt und nur Administratoren können sich am System anmelden.
 - Eine Übersicht informiert Administratoren über aktive Sitzungen, welche gesammelt über einen Klick oder einzeln beendet werden können.
- Möglichkeit hinzugefügt um den SysConfig-Import über die Konfiguration zu deaktivieren
- Added Apache MD5 as a new password hashing backend, thanks to Norihiro Tanaka.
- Added the possibility to restrict customer self registration by email address whitelist or blacklist, thanks to Renée Bäcker.
- Neues Übersichtsseitenmodul hinzugefügt, um die Ausgabe eines externen Kommandos auszugeben, Danke an ib.pl

2.2.6.5. Entwicklung

- Neue Template-Engine basierend auf Template::Toolkit.
- Ein zentraler Objekt-Manager erleichtert das Erstellen und Nutzen von globalen Objekten. (Danke an Moritz Lenz von noris network).
- The OPM package format was extended to signal that a package has been merged into another package, allowing the package manager to correctly handle this situation on package installation or update.

- Das Caching wurde in einem globalen Cache-Objekt zentralisiert, das ebenfalls die gesamten Daten cached.
- Cache Benchmark Skript wurde hinzugefügt, Danke an ib.pl

2.2.7. New Features of OTRS 3.3

2.2.7.1. Produktivität

- Dashboard ticket lists and regular ticket overviews can now be filtered by eligible ticket columns, and the shown columns are configurable.
- Die medium- und Vorschauansicht der Ticketübersichten sind jetzt sortierbar.
- Es wurde ein Kalender-Widget zur Übersichtsseite hinzugefügt, das Tickets als Ereignis anzeigen kann.
- Es wurde ein Widget zur Übersichtsseite hinzugefügt, das Tickets pro Status und Warteschlange in Form einer Matrize anzeigen kann.
- Agenten können nun wichtige Artikel markieren.
- Ein neues Baumauswahl-Widget erleichtert und beschleunigt die Arbeit mit verzweigten Daten (Queues, Services, etc.).
- Unterstützung für die Suche nach relativen Zeiträumen (z.B. "vor mehr als einem Monat") für dynamische Datums- und Datum-Zeit-Felder hinzugefügt.
- Es ist nun möglich, Vorlagen (vorher "Standard-Antworten") auch für die Erstellung neuer oder Weiterleitung bestehender Tickets zu definieren.
- Die Liste verfügbarer Prozesse kann nun per ACL gefiltert werden.
- Starten von Prozessen nun auch über die Kunden-Oberfläche möglich.
- An vielen Stellen wird Text nun nicht mehr nach einer festen Anzahl von Zeichen abgeschnitten, sondern abhängig vom verfügbaren Platz auf dem Bildschirm. So wird mehr Information sichtbar dargestellt.
- OTRS ist jetzt für Retina-Bildschirme geeignet. Bilder wurden für höhere Auflösungen angepasst und viele Symbole durch Schriftzeichen aus der FontAwesome-Schriftart ersetzt.
- Mithilfe der neuen "Verwaltungsübersichtsseite,-"Funktion ist es jetzt möglich, grafische Statistiken (z.B. als Balken- oder Linien-Diagramm) in der Übersichtsseite anzuzeigen. Hinweis: Diese Funktion wird vom IE8 nicht unterstützt.

2.2.7.2. Arbeiten mit externen Systemen

- OTRS kann nun mehrere Kundenfirmen-Datenbanken nutzen (Danke an Cyrille @ belnet-ict).
- OTRS kann nun Kundendaten automatisch (permanent) in dynamischen Feldern speichern. Dies ist hilfreich für Reportingzwecke.
- OTRS kann nun eingehende E-Mails korrekt bestehenden Tickets anhand von externen Ticketnummern zuordnen.
- OTRS kann nun auch E-Mails über POP3/TLS-Verbindungen abholen.

2.2.7.3. Installation & Administration

- Der Web-Installer kann OTRS nun (zusätzlich zu MySQL) auch auf PostgreSQL-, Oracle- und SQL-Server-Datenbanken installieren.
- OTRS unterstützt MySQL 5.6 nun vollständig.
- Generic Agent-Aufgaben können jetzt bei konfigurierten Ticket-Ereignissen ausgeführt werden.
- Der neue grafische ACL-Editor erleichtert das Bearbeiten von Zugriffskontrolllisten (ACLs).
- Postmaster-Filter können jetzt negierte Filterbedingungen verwenden (Danke an Renée Bäcker).
- Postmaster-Filter können jetzt relative "Warten bis"-Zeiten und Besitzer/Verantwortlichen für neue Tickets auf Basis von eingehenden E-Mail-Daten festlegen.
- Kunden- und Agentenpasswörter können jetzt mit dem (im Vergleich zu SHA) stärkeren bcrypt-Algorithmus verschlüsselt werden.
- Viele Symbole wurden durch Schriftzeichen ersetzt, was das Erstellen eigener Skins mit angepasster Farbgebung stark erleichtert und zudem die Ladezeiten verbessert.

2.3. Hardware and Software Requirements

OTRS can be installed on many different operating systems. OTRS can run on linux and on other unix derivates (e.g. OpenBSD or FreeBSD). 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.

Um OTRS zu betreiben, wird ein Web- sowie ein Datenbankserver benötigt. Weiterhin ist eine funktionierende Perl-Installation mit einigen Zusatzmodulen die Grundvoraussetzung für ein funktionierendes System. Der Webserver und Perl müssen auf der gleichen Maschine installiert sein, auf der später auch OTRS ausgeführt werden soll. Das Datenbank-Back-End kann auf der lokalen oder auf einer entfernten Maschine installiert werden.

Für den Webserver empfehlen wir die Verwendung von apache 2. Durch die Verwendung des Modules mod_perl, kann die Geschwindigkeit von OTRS enorm gesteigert werden. Prinzipiell sollte OTRS aber auf jedem Webserver betrieben werden können, der die Ausführung von Perl-Skripten unterstützt.

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

Für Perl benötigen Sie einige Zusatzmodule, die Sie entweder direkt über die Shell von Perl und CPAN oder mit Hilfe des Paketmanagers Ihres Betriebssystems installieren können (rpm, yast, apt-get).

Software-Anforderungen

2.3.1. PERL Unterstützung

- Perl 5.10 oder höher

2.3.2. Web Server Unterstützung

- Apache2 + mod_perl2 oder höher (empfohlen)

- Webserver mit CGI support (CGI nicht empfohlen)

2.3.3. Datenbankunterstützung

- MySQL 5.0 oder höher
- MariaDB
- PostgreSQL 8.4 oder höher
- Oracle 10g oder höher

Im Abschnitt für die Installation der für OTRS benötigten Perl-Module wird beschrieben, wie Sie Perl-Module manuell einspielen können.

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

2.3.4. Webbrowser-Unterstützung

Um OTRS zu benutzen benötigen Sie einen modernen Browser mit eingeschaltetem JavaScript Support. Die folgenden Browser werden nicht unterstützt:

- Internet Explorer vor Version 10
- Firefox vor Version 10
- Safari vor Version 5

We recommend keeping your browser up-to-date. JavaScript and rendering performance in newer versions is always improved. Dramatic performance issues can be seen in larger systems when using older versions. We are happy to consult you on that matter.

2.4. Community

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

Die Webseite der OTRS-Community ist: <http://www.otrs.com/open-source/>.

2.5. Kommerzieller Support und Dienstleistungen für OTRS

Our [OTRS Business Solution™](#) offers you best professional support from the OTRS team, reliable OTRS security and regular free updates as well as an [exclusive set of additional Business Features](#) that you can flexibly activate or deactivate according to different deployment scenarios.

Die [OTRS Group](#) bietet [Trainingsprogramme](#) in verschiedenen Ländern an. Sie können entweder an einem unserer regelmäßig stattfindenden öffentlichen OTRS Administrator-Trainings teilnehmen oder an einer Inhouse-Schulung, die alle spezifischen Anforderungen Ihres Unternehmens abdeckt.

Kapitel 2. Installation

This chapter describes the installation and basic configuration of the central OTRS framework. It covers information on installing OTRS from source, or with a binary package such as an RPM.

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. The Simple Way - Installation of Pre-Built Packages

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

1.1. Installation des RPM auf einem SuSE-Linux-Server

This section describes the installation of our RPM package on a SUSE Linux server.

1.1.1. Vorbereiten der Datenbank für OTRS

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

Anmerkung

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

Installieren Sie MySQL als root mit folgenden Befehl:

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

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

```
max_allowed_packet = 20M
query_cache_size = 32M
innodb_log_file_size = 256M
```

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

1.1.2. OTRS installieren

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

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

Now restart Apache with the command **systemctl restart apache2.service** to load the configuration changes for OTRS.

1.1.3. Installation zusätzlicher Perl-Module

OTRS needs some more modules than can be installed by the RPM. You can post-install them manually. You can check what modules you are missing by running the `bin/otrs.CheckModules.pl` script located in the `/opt/otrs` directory. Some modules are only needed for optional functionality, such as communication with IMAP(S) servers or PDF generation. On SLES you should add an external repository to the zypper configuration in order to get the modules needed for your system. Choose the module needed for your OS version from here: <http://download.opensuse.org/repositories/devel:/languages:/perl/>. Add the repository like this for SLES 11 SP2:

```
zypper ar -f -n perl http://download.opensuse.org/repositories/devel:/languages:/perl/
SLE_11_SP2_Perl
```

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

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

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

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

```
Key ID: DCCA98DDCECF338C
Key Name: devel:languages:perl OBS Project &lt;devel:languages:perl@build.opensuse.org&gt;
Key Fingerprint: 36F0AC0BCA9D8AF2871703C5DCCA98DDCECF338C
Key Created: Wed Oct 10 22:04:18 2012
Key Expires: Fri Dec 19 22:04:18 2014
Repository: perl
```

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

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

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

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

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

Now you can start the OTRS daemon and activate corresponding watchdog cron job (this must be done by the otrs user):

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

That's it, congratulations!

1.2. Installation von OTRS auf einem REd Hat Enterprise Linux oder CentOS-System

This section describes the installation of our RPM package on a Red Hat Enterprise Linux (RHEL) or CentOS server. For OTRS 3.3 and up, RHEL 6 or CentOS 6 is a requirement, version 5 is not supported.

1.2.1. Vorbereiten der Datenbank für OTRS

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

Installieren Sie MySQL als root mit folgenden Befehl:

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

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

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



```
innodb_log_file_size = 256M
```

Now execute **systemctl start mysqld** to re-start the database server and activate these changes. Then run **/usr/bin/mysqld_secure_installation** and follow the on-screen instructions to set a database root password, remove anonymous access and remove the test database.

1.2.2. OTRS installieren

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

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

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

```
Transaction Summary
-----
Install      26 Package(s)

Total size: 80 M
Total download size: 6.0 M
Installed size: 88 M
Downloading Packages:
(1/25): apr-1.3.9-5.el6_2.x86_64.rpm | 123 kB | 00:00
...
(25/25): procmail-3.22-25.1.el6.x86_64.rpm | 163 kB | 00:00
-----
Total                               887 kB/s | 6.0 MB | 00:06
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : apr-1.3.9-5.el6_2.x86_64 | 1/26
  ...
  Installing : otrs-x.x.x-01.noarch | 26/26
Check OTRS user ... otrs added.
...
[root@otrs-centos6 ~]#
```

Now restart Apache with the command **systemctl restart httpd.service** to load the configuration changes for OTRS.

1.2.3. Installation zusätzlicher Perl-Module

OTRS needs some more modules than can be installed by the RPM. You can post-install them manually. You can check what modules you are missing by running the `bin/otrs.CheckModules.pl` script located in the `/opt/otrs` directory. Some modules are only needed for optional functionality, such as communication with IMAP(S) servers or PDF generation. On Red Hat or CentOS we recommend installing these modules from the EPEL

repository, a repository maintained by the Fedora project, which provides high quality packages for RHEL and derivatives. Check for more information [the EPEL web site](http://www.fedoraproject.org/wiki/Extra_packages_for_RHEL).

If you're on RHEL 6 or CentOS 6, you can get the latest package for EPEL from [this site](http://www.fedoraproject.org/wiki/Extra_packages_for_RHEL). You can add this repository to yum it in one go by copying the RPM URL you find on this page and executing this command:

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

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
epel-release noarch 6-8 /epel-release-6-8.noarch 22 k
Transaction Summary
=====
Install 1 Package(s)

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

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

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

Wenn Sie yum das erste Mal ausgeführt, nachdem Sie das Repository hinzugefügt haben, werden Sie aufgefordert die Repository-Schlüssel hinzuzufügen. Nun können Sie fehlende Module, wie folgt installieren.

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

Dependencies Resolved

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

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

Total download size: 71 k
Installed size: 154 k
Downloading Packages:
perl-Text-CSV_XS-0.85-1.el6.x86_64.rpm                | 71 kB    00:00
warning: rpmts_HdrFromFdno: Header V3 RSA/SHA256 Signature, key ID 0608b895: NOKEY
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6
Importing GPG key 0x0608B895:
  Userid : EPEL (6) <epel@fedoraproject.org>;
  Package: epel-release-6-8.noarch (@/epel-release-6-8.noarch)
  From   : /etc/pki/rpm-gpg/RPM-GPG-KEY-EPEL-6
Is this ok [y/N]: y
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : perl-Text-CSV_XS-0.85-1.el6.x86_64      1/1
  Verifying  : perl-Text-CSV_XS-0.85-1.el6.x86_64      1/1

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

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

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

Now you can start the OTRS daemon and activate corresponding watchdog cron job (this must be done by the otrs user):

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

That's it, congratulations!

1.2.4. Installation der Oracle-Datenbanktreiber auf Red Hat / CentOS

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

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

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

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

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

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

```
[root@otrs-centos6 otrs]# export ORACLE_HOME=/usr/lib/oracle/11.2/client64
[root@otrs-centos6 otrs]# export LD_LIBRARY_PATH=$ORACLE_HOME/lib
[root@otrs-centos6 otrs]# cpan
cpan[1]> look DBD::Oracle
...
Fetching with LWP:
  http://www.perl.org/CPAN/authors/id/P/PY/PYTHIAN/CHECKSUMS
Checksum for /root/.cpan/sources/authors/id/P/PY/PYTHIAN/DBD-Oracle-1.62.tar.gz ok
Scanning cache /root/.cpan/build for sizes
DONE
...
Working directory is /root/.cpan/build/DBD-Oracle-1.62-ZH6LNy
[root@localhost DBD-Oracle-1.62-ZH6LNy]# perl Makefile.PL
...
[root@localhost DBD-Oracle-1.62-ZH6LNy]# make
...
[root@localhost DBD-Oracle-1.62-ZH6LNy]# make install
...
cpan[2]> exit
Terminal does not support GetHistory.
Lockfile removed.
```

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

1.3. Installation von OTRS auf einem Debian- oder Ubuntu-System

Wichtig

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

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

```
apt-get install libapache2-mod-perl2 libdbd-mysql-perl libtimedate-perl libnet-dns-perl
libnet-ldap-perl \
libio-socket-ssl-perl libpdf-api2-perl libdbd-mysql-perl libsoap-lite-perl libtext-csv-
xs-perl \
libjson-xs-perl libapache-dbi-perl libxml-libxml-perl libxml-libxslt-perl libyaml-perl \
libarchive-zip-perl libcrypt-eksblowfish-perl libencode-hanextra-perl libmail-
imapclient-perl \
libtemplate-perl
```

2. Installation From Source (Linux, Unix)

2.1. Schritt 1: Installation des .tar.gz

If you want to install OTRS from source, first download the source archive as .tar.gz, .tar.bz2, or .zip file from <http://www.otrs.com/try/>

Entpacken Sie das Archiv (zum Beispiel mit **tar**) in das Verzeichnis /opt und benennen Sie das entpackte Verzeichnis von 'otrs-x.x.x' in 'otrs' um (siehe unteres Beispiel)

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

2.2. Schritt 2: Installation zusätzlicher Perl-Module

Use the following script to get an overview of all installed and required CPAN modules.

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

Anmerkung

Please note that OTRS requires a working Perl installation with all "core" modules such as the module version. These modules are not explicitly checked by the script. You may need to install a perl-core package on some systems like RHEL that do not install the Perl core packages by default.

To install missing Perl modules, you can:

2.2.1. a) Installieren Sie die Pakete über den Paketmanager Ihrer Linuxdistribution

- For Red Hat, CentOS, Fedora or compatible systems:

```
shell> yum install "perl(Digest::MD5)"
```

- For SUSE Linux Enterprise Server, openSUSE or compatible systems: first determine the name of the package the module is shipped in. Usually the package for My::Module would be called "perl-My-Module".

```
shell> zypper search Digest::MD5
```

Then install:

```
shell> zypper install perl-Digest-MD5
```

- For Debian, Ubuntu or compatible systems first determine the name of the package the module is shipped in. Usually the package for My::Module would be called "libmy-module-perl".

```
shell> apt-cache search Digest::MD5
```

Then install:

```
shell> apt-get install libdigest-md5-perl
```

Please note that it might be that you can't find all modules or their required versions in your distribution repository, in that case you might choose to install those modules via CPAN (see below).

2.2.2. b) Installieren Sie die benötigten Module über die Shell mit CPAN

Note that when you're on Linux you should run CPAN as your superuser account because the modules should be accessible both by the OTRS account and the account under which the web server is running.

```
shell> perl -MCPAN -e shell;  
...  
install Digest::MD5  
install Crypt::PasswdMD5  
...
```

Any optional modules listed by the script should be installed depending on the special requirements of the target system.

2.3. Schritt 3: OTRS-Nutzer erstellen

Create user:

```
shell> useradd -d /opt/otrs/ -c 'OTRS user' otrs
```

Add user to webserver group (if the webserver is not running as the OTRS user):

```
shell> usermod -G www otrs  
(SUSE=www, Red Hat/CentOS/Fedora=apache, Debian/Ubuntu=www-data)
```

2.4. Schritt 4: Aktivieren der Standard-Konfigurationsdateien

There are two OTRS config files bundled in `$OTRS_HOME/Kernel/*.dist` and `$OTRS_HOME/Kernel/Config/*.dist`. You must activate them by copying them without the ".dist" filename extension.

```
shell> cd /opt/otrs/  
shell> cp Kernel/Config.pm.dist Kernel/Config.pm
```

2.5. Schritt 5: Überprüfen Sie ob alle benötigten Module installiert sind

```
shell> perl -cw /opt/otrs/bin/cgi-bin/index.pl
/opt/otrs/bin/cgi-bin/index.pl syntax OK

shell> perl -cw /opt/otrs/bin/cgi-bin/customer.pl
/opt/otrs/bin/cgi-bin/customer.pl syntax OK

shell> perl -cw /opt/otrs/bin/otrs.Console.pl
/opt/otrs/bin/otrs.Console.pl syntax OK
```

"syntax OK" tells you all mandatory Perl modules are installed.

2.6. Schritt 6: Konfiguration des Apache-Web-servers

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

```
# rhel / centos:
shell> yum install httpd mod_perl

# suse:
shell> zypper install apache2-mod_perl

# debian/ubuntu:
shell> apt-get install apache2 libapache2-mod-perl2
```

Most Apache installations have a conf.d directory included. On Linux systems you can usually find this directory under /etc/apache or /etc/apache2. Log in as root, change to the conf.d directory and link the appropriate template in /opt/otrs/scripts/apache2-httpd.include.conf to a file called zzz_otrs.conf in the Apache configuration directory (to make sure it is loaded after the other configurations).

OTRS requires a few Apache modules to be active for optimal operation. On most platforms you can make sure they are active via the tool a2enmod.

```
shell> a2enmod perl
shell> a2enmod version
shell> a2enmod deflate
shell> a2enmod filter
shell> a2enmod headers
```

Now you can restart your web server to load the new configuration settings. On most systems you can do that with the command **systemctl restart apache2.service**.

2.7. Schritt 7: Dateiberechtigungen

File permissions need to be adjusted to allow OTRS to read and write files:

```
otrs.SetPermissions.pl [ --otrs-user= OTRS user, defaults to 'otrs' ] { --web-group= group
of the web server user }
```

For example:

- Web server which runs as the OTRS user:

```
shell> bin/otrs.SetPermissions.pl --web-user=otrs
```

- Webserver with wwwrun user (e. g. SUSE):

```
shell> bin/otrs.SetPermissions.pl --web-group=wwwrun
```

- Webserver with apache user (e. g. Red Hat, CentOS):

```
shell> bin/otrs.SetPermissions.pl --web-group=apache
```

- Webserver with www-data user (e. g. Debian, Ubuntu):

```
shell> bin/otrs.SetPermissions.pl --web-group=www-data
```

2.8. Schritt 8: Datenbankeinrichtung und grundlegende Systemkonfiguration

Please use the web installer at <http://yourhost/otrs/installer.pl> (replace "yourhost" with your OTRS hostname) to setup your database and basic system settings such as email accounts.

Anmerkung

The following configuration settings are recommended for MySQL setups. Please add the following lines to `/etc/my.cnf` under the `[mysqld]` section:

```
max_allowed_packet = 20M
query_cache_size = 32M
innodb_log_file_size = 256M
```

2.9. Schritt 9: Erste Anmeldung

Now you are ready to login to your system at <http://yourhost/otrs/index.pl> with the credentials you configured in the web installer (User: root@localhost).

Nach diesem Schritt ist das Basis-Systemsetup beendet.

2.10. Step 10: Start the OTRS Daemon

The new OTRS daemon is responsible for handling any asynchronous and recurring tasks in OTRS. What has been in cron file definitions previously is now handled by the OTRS daemon, which is now required to operate OTRS. The daemon also handles all GenericAgent jobs and must be started from the otrs user.

```
shell> /opt/otrs/bin/otrs.Daemon.pl start
```

2.11. Step 11: Cron jobs for the OTRS user

There are two default OTRS cron files in `/opt/otrs/var/cron/*.dist`, and their purpose is to make sure that the OTRS Daemon is running. They need to be activated by copying them without the ".dist" filename extension.


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

To schedule these cron jobs on your system, you can use the script `Cron.sh` with the `otrs` user.

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

Stopping the cron jobs is also possible (useful for maintenance):

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

2.12. Step 12: Setup bash autocompletion (optional)

All regular OTRS commandline operations happen via the `otrs Console` interface `bin/otrs.Console.pl`. This provides an auto completion for the bash shell which makes finding the right command and options much easier.

You can activate the bash autocompletion by installing the package `bash-completion`. It will automatically detect and load the file `/opt/otrs/.bash_completion` for the `otrs` user.

After restarting your shell, you can just type `bin/otrs.Console.pl` followed by `TAB`, and it will list all available commands. If you type a few characters of the command name, `TAB` will show all matching commands. After typing a complete command, all possible options and arguments will be shown by pressing `TAB`.

2.13. Step 13: Further Information

We advise you to read the OTRS performance tuning chapter.

If you encounter problems with the installation, you can send a message to our mailing list `otrs@otrs.org` (<http://lists.otrs.org/>).

You can also ask the OTRS Group to either help you in planning or deploying OTRS, or review your installed OTRS system. Our [professional services](#) are designed to help you deploy OTRS faster and to get the most benefit out of OTRS.

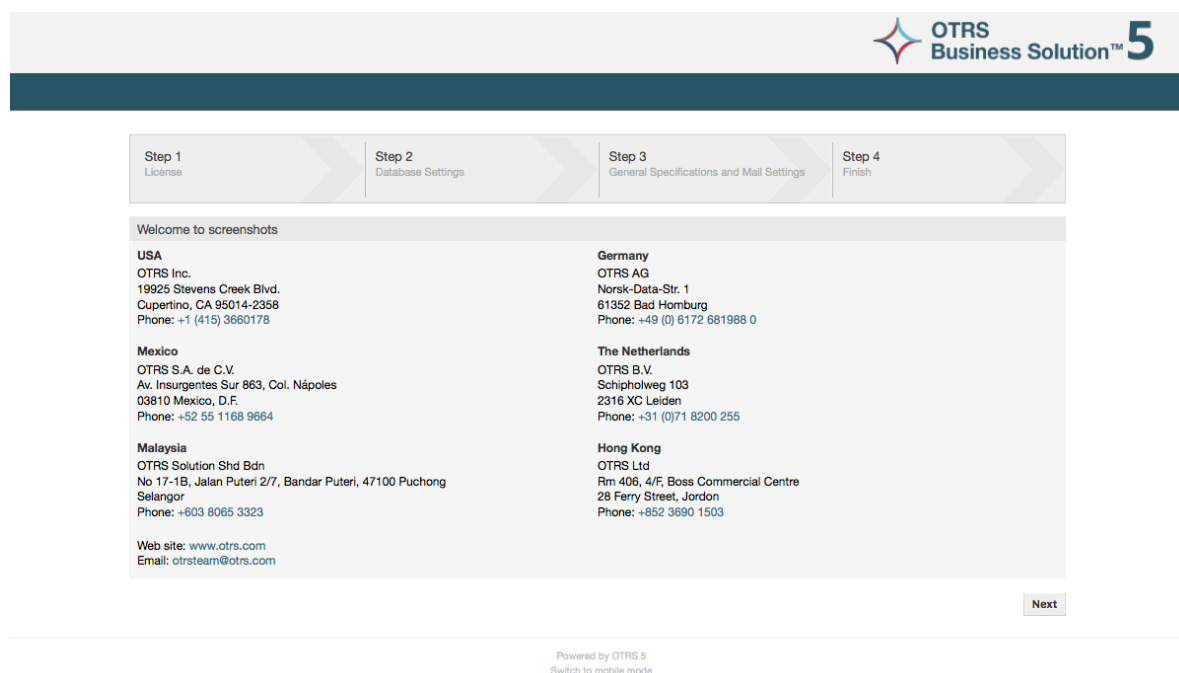
3. Using the Web Installer

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

Wenn der Webinstaller gestartet ist, folgen Sie den Anweisungen auf dem Bildschirm.

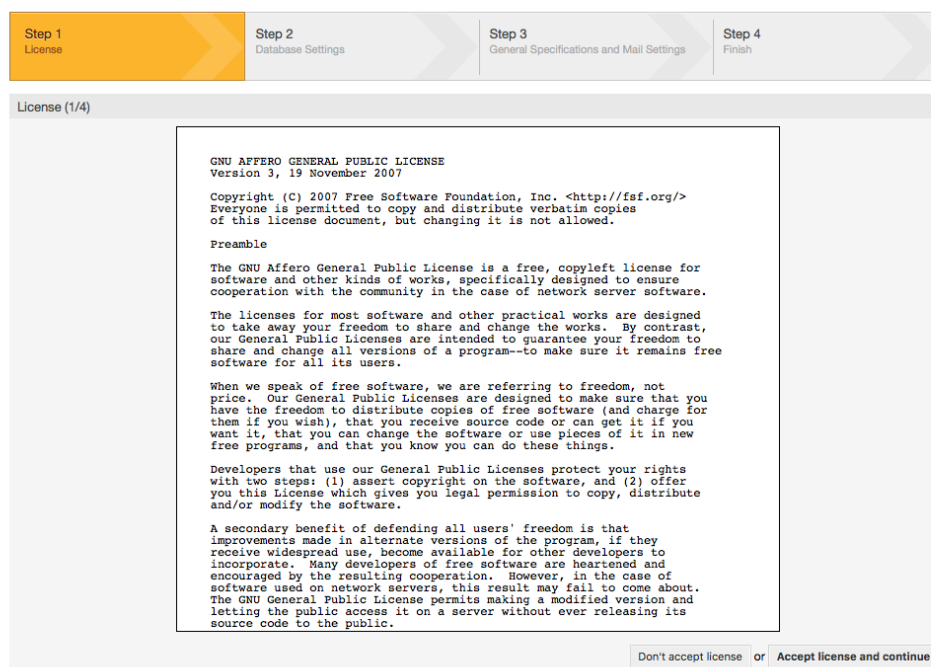
1. Check out the information about the OTRS offices and click on 'Next' to continue (see figure below).

Abbildung 2.1. Welcome screen



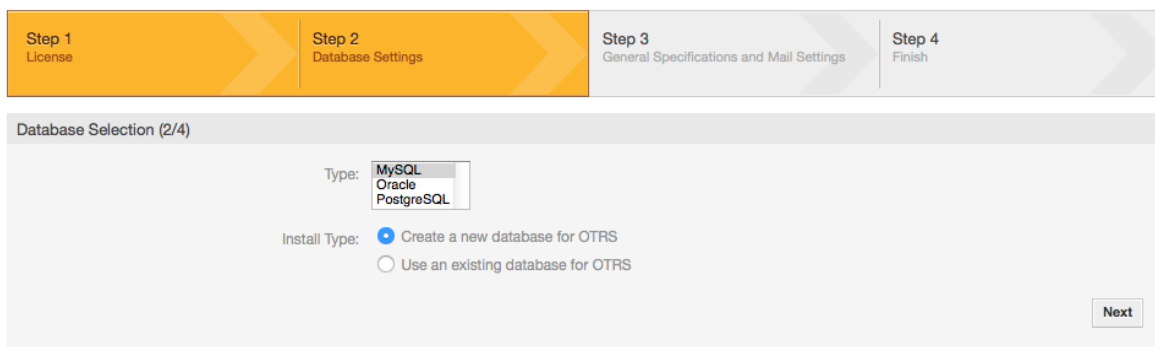
2. Read the GNU Affero General Public License (see figure below) and accept it, by clicking the corresponding button at the bottom of the page.

Abbildung 2.2. GNU Affero General Public License



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

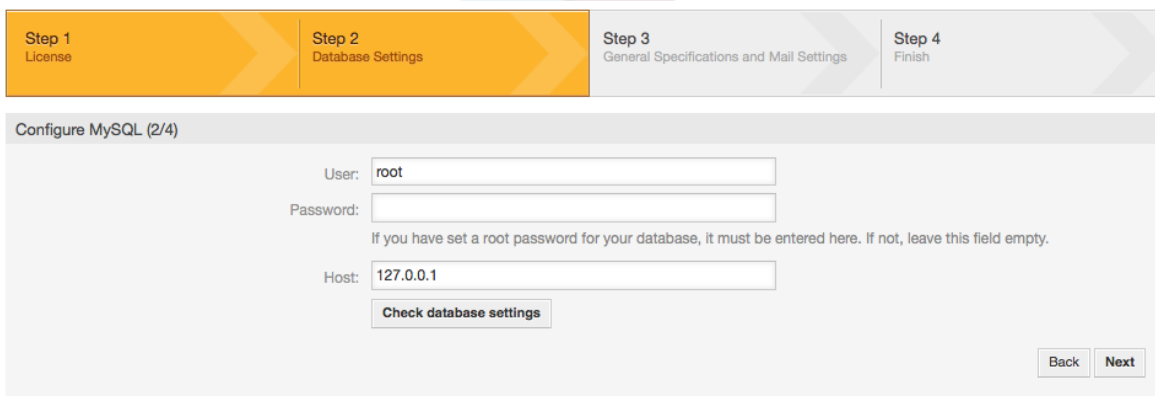
Abbildung 2.3. Datenbank-Auswahl



The screenshot shows the 'Database Selection (2/4)' screen. At the top, there is a progress bar with four steps: Step 1 License, Step 2 Database Settings (highlighted in orange), Step 3 General Specifications and Mail Settings, and Step 4 Finish. Below the progress bar, the 'Type:' dropdown menu is set to 'MySQL', with 'Oracle' and 'PostgreSQL' as options. Under 'Install Type:', the radio button for 'Create a new database for OTRS' is selected, while 'Use an existing database for OTRS' is unselected. A 'Next' button is located in the bottom right corner.

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

Abbildung 2.4. Database credentials



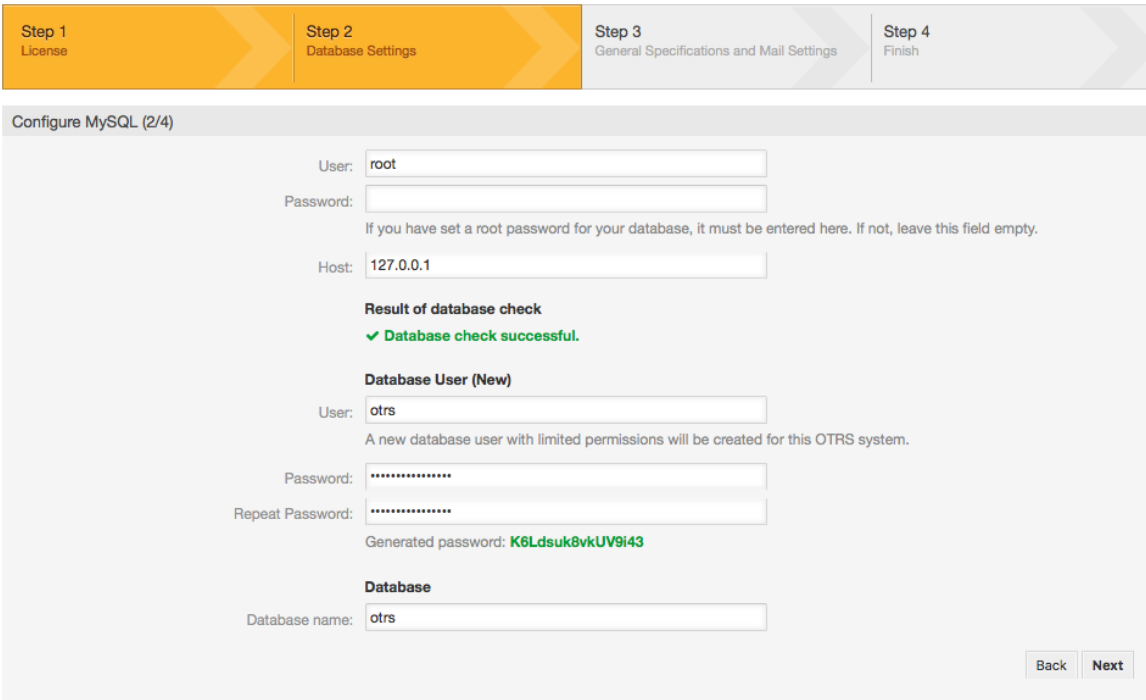
The screenshot shows the 'Configure MySQL (2/4)' screen. At the top, there is a progress bar with four steps: Step 1 License, Step 2 Database Settings (highlighted in orange), Step 3 General Specifications and Mail Settings, and Step 4 Finish. Below the progress bar, the 'User:' field contains 'root'. The 'Password:' field is empty. Below the password field, there is a note: 'If you have set a root password for your database, it must be entered here. If not, leave this field empty.' The 'Host:' field contains '127.0.0.1'. A 'Check database settings' button is located below the host field. In the bottom right corner, there are 'Back' and 'Next' buttons.

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

Warnung

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

Abbildung 2.5. Database settings



Step 1 License | **Step 2 Database Settings** | Step 3 General Specifications and Mail Settings | Step 4 Finish

Configure MySQL (2/4)

User:

Password:

If you have set a root password for your database, it must be entered here. If not, leave this field empty.

Host:

Result of database check
 ✓ Database check successful.

Database User (New)

User:

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

Password:

Repeat Password:

Generated password: **K6Ldsuk8vkUV9i43**

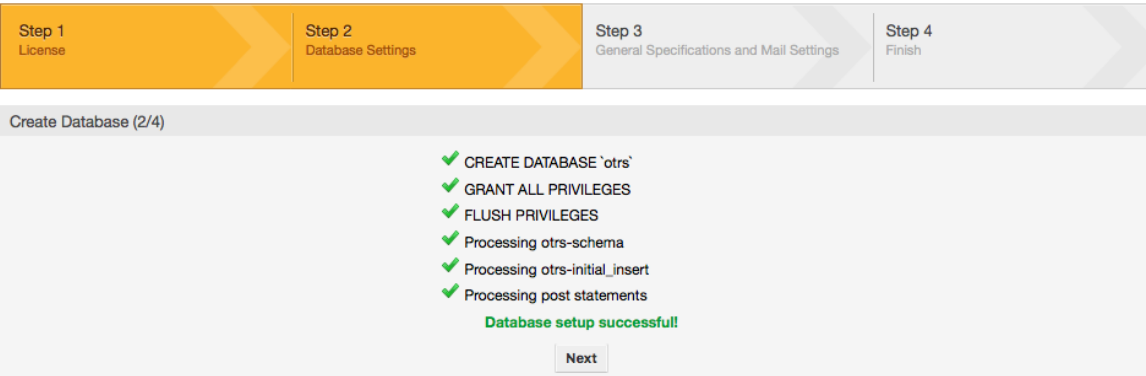
Database

Database name:

Back Next

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

Abbildung 2.6. Successful database setup



Step 1 License | **Step 2 Database Settings** | Step 3 General Specifications and Mail Settings | Step 4 Finish

Create Database (2/4)

- ✓ CREATE DATABASE `otrs`
- ✓ GRANT ALL PRIVILEGES
- ✓ FLUSH PRIVILEGES
- ✓ Processing otrs-schema
- ✓ Processing otrs-initial_insert
- ✓ Processing post statements

Database setup successful!

Next

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

Abbildung 2.7. System settings

Step 1
License
Step 2
Database Settings
Step 3
General Specifications and Mail Settings
Step 4
Finish

System Settings (3/4)

SystemID: The identifier of the system. Each ticket number and each HTTP session ID contain this number.

System FQDN: Fully qualified domain name of your system.

AdminEmail: Email address of the system administrator.

Organization:

Log

LogModule: Log backend to use.

LogFile:

Webfrontend

Default language: Default language.

CheckMXRecord: Email addresses that are manually entered are checked against the MX records found in DNS. Don't use this option if your DNS is slow or does not resolve public addresses.

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

Abbildung 2.8. Mail configuration

Step 1
License
Step 2
Database Settings
Step 3
General Specifications and Mail Settings
Step 4
Finish

Mail Configuration (3/4)

Configure Outbound Mail

Outbound mail type: Select outbound mail type.

Outbound mail port: Select outbound mail port.

Configure Inbound Mail

Inbound mail type: Select inbound mail type.

Inbound mail host: Inbound mail host.

Inbound mail user: User for inbound mail.

Inbound mail password: Password for inbound mail.

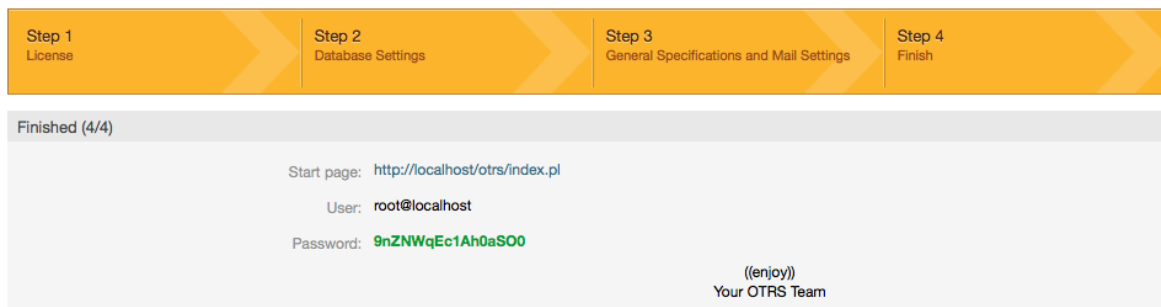
or

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

Warnung

Bitte notieren Sie das generierte Passwort für den Benutzer 'root@localhost'.

Abbildung 2.9. Web installer final screen



4. OTRS auf Windows

OTRS kann auf einer Vielzahl von Systemplattformen betrieben werden kann, darunter Enterprise Linux, wie z. B. Red Hat Enterprise Linux, oder SuSE Linux Enterprise Server, sowie auf einer Vielzahl weiterer Linux-Derivate.

Beim Betrieb von OTRS auf Windows-Plattformen mussten wir jedoch immer wieder starke Performance-Einbußen feststellen. Diese konnten auch nach langer und intensiver Analyse aufgrund technischer Differenzen nicht zu unserer Zufriedenheit gelöst werden. Die Entwicklung unseres Windows-Installers und der OTRS Appliance haben wir aufgrund einer nun nur noch eingeschränkten Verfügbarkeit benötigter Drittkomponenten durch andere Hersteller eingestellt.

Unter diesen Bedingungen können wir keinen zukunftssicheren Betrieb von OTRS auf Windows-Plattformen garantieren und empfehlen daher die Migration auf eine der oben genannten Linux-Plattformen oder den Einsatz unserer **OTRS Business Solution™ Managed**.

Wir möchten Ihnen die Migration von Windows auf Linux erleichtern und Ihnen damit die beste Performance von OTRS bieten. Daher haben wir hier eine detaillierte Anleitung für Sie zusammengestellt.

4.1. Migration existierender Windows-Installationen nach Linux

4.1.1. Einführung und Vorbereitung

If you have a Windows based installation and you would like to change to a Linux based system you will need to setup a Linux server or virtual machine and install OTRS there (see the installation instructions). This will be the target system for the migration.

4.1.2. Downloaden Sie das OTRSCloneDB-Skript um Datenbanken zu klonen

Please go to the admin menu of the Windows based system and install the newest version of OTRSCloneDB package into your OTRS:

Sie können das OTRSCloneDB-Paket direkt aus dem Paket-Manager installieren. Wählen Sie "OTRS Extensions" aus der Dropdown-Liste auf der linken Seite aus und klicken Sie

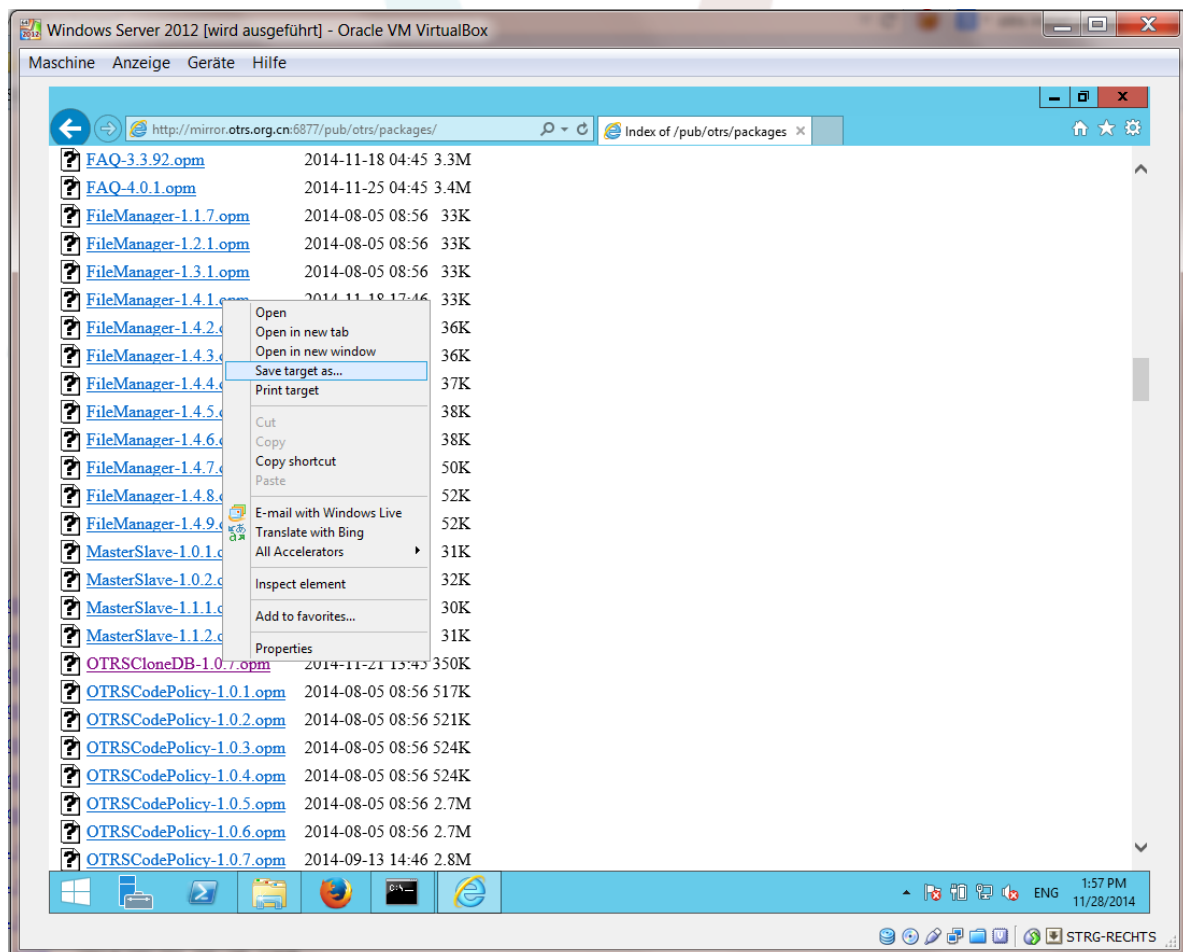
auf den Button "Verzeichnis aktualisieren". Das Paket OTRSCloneDB wird dann in einer Liste angezeigt, klicken Sie dann auf "Installieren".

You could also download the package manually from the OTRS FTP server and install the package manually as described below. <http://ftp.otrs.org/pub/otrs/packages/>

Downloaden Sie das Paket mit der höchsten Versionsnummer:

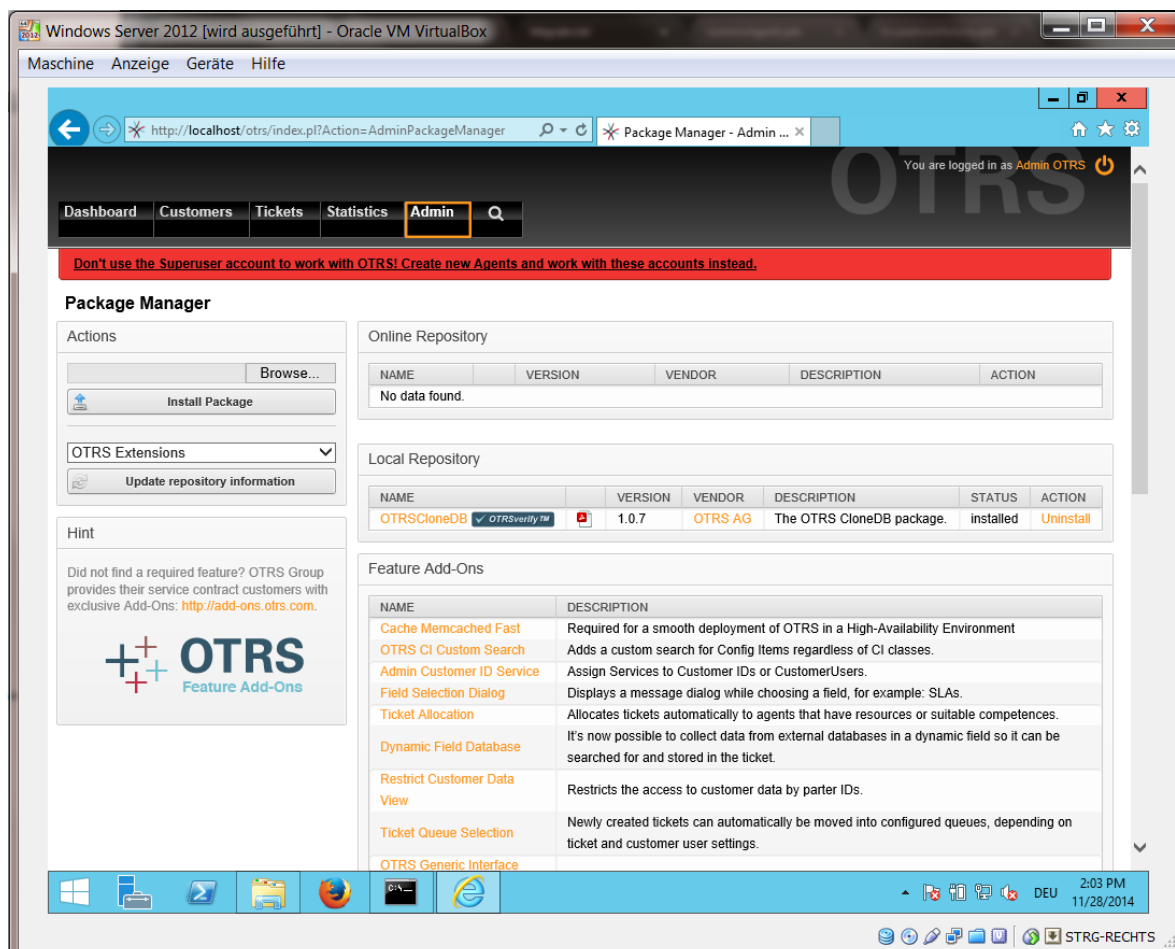
OTRSCloneDB-1.0.13.opm

Abbildung 2.10. Download des OTRSCloneDB-Skript - Screenshot



and install it to your Windows based installation:

Abbildung 2.11. Installieren des OTRSCloneDB-Skripts - Screenshot



It is also no problem if you have installed some additional features or custom developments on your OTRS. You just need to take care that all of your installed packages are also compatible with Unix based systems. For packages provided by OTRS this is the case.

4.1.3. Enable remote access for the PostgreSQL database of target system

The OTRSCloneDB script will copy the database data over the network, so we need to enable remote access to the database. The setup is different for the different databases, we will describe opening remote access for a PostgreSQL database here.

After logging into your target system via SSH you need to change into the postgresql directory:

```
Shell> cd /etc/postgresql/9.4/main
Shell> vi postgresql.conf
```

Fügen Sie folgende Zeile am Ende der Datei hinzu:

```
listen_addresses = '*'
```

Speichern Sie die Datei.


```
Shell> vi pg_hba.conf
```

Fügen Sie folgende Zeile am Ende der Datei hinzu:

```
host all all 0.0.0.0/0 md5
```

Speichern Sie die Datei.

Restart your PostgreSQL server:

```
Shell> service postgresql restart
```

4.1.4. OTRS-Dienste stoppen

Stop all running services of your target system:

```
shell> service cron stop
shell> service apache2 stop
shell> su - otrs
shell> cd /opt/otrs/
shell> bin/Cron.sh stop
shell> bin/otrs.Daemon.pl -a stop
shell> exit
```

4.1.5. Drop the existing database of your target system to have an empty database for the clone data

The OTRSCloneDB script will not remove the data in the existing otrs database of the target system, so we need to do this manually:

Change the user to the postgresql user:

```
Shell> su - postgres
```

Drop the existing otrs database:

```
Shell> dropdb otrs
```

Create a new otrs database for the otrs user:

```
Shell> createdb --owner=otrs --encoding=utf8 otrs
```

Go back to root user:

```
Shell> exit
```

4.1.6. Get the PostgreSQL password of your database

Change into the OTRS directory of your target system:

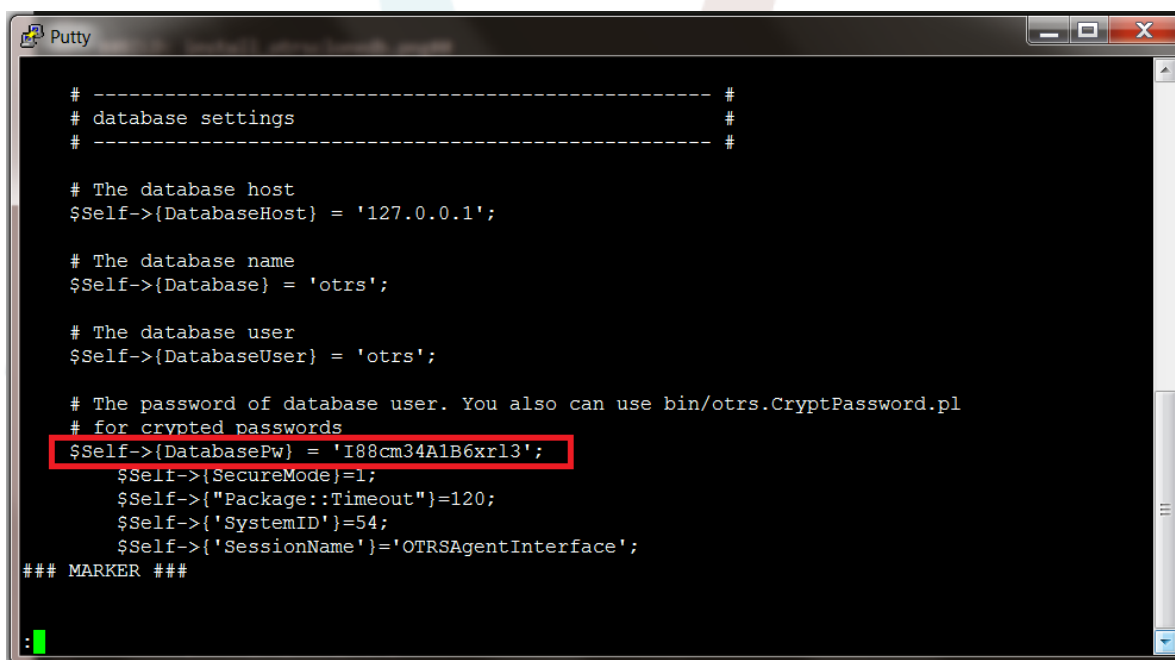
```
Shell> cd /opt/otrs
```

and take a look at the configuration file of your target system:

```
Shell> less Kernel/Config.pm
```

Sie finden das Datenbank-Passwort, wenn Sie ein wenig scrollen:

Abbildung 2.12. Zieldatenbankpasswort holen - Bildschirmfoto



```
Putty
# ----- #
# database settings #
# ----- #

# The database host
$Self->{DatabaseHost} = '127.0.0.1';

# The database name
$Self->{Database} = 'otrs';

# The database user
$Self->{DatabaseUser} = 'otrs';

# The password of database user. You also can use bin/otrs.CryptPassword.pl
# for crypted passwords
$Self->{DatabasePw} = 'I88cm34A1B6xr13';
$Self->{SecureMode}=1;
$Self->{"Package::Timeout"}=120;
$Self->{'SystemID'}=54;
$Self->{'SessionName'}='OTRSagentInterface';
### MARKER ###
```

In unserem Beispiel:

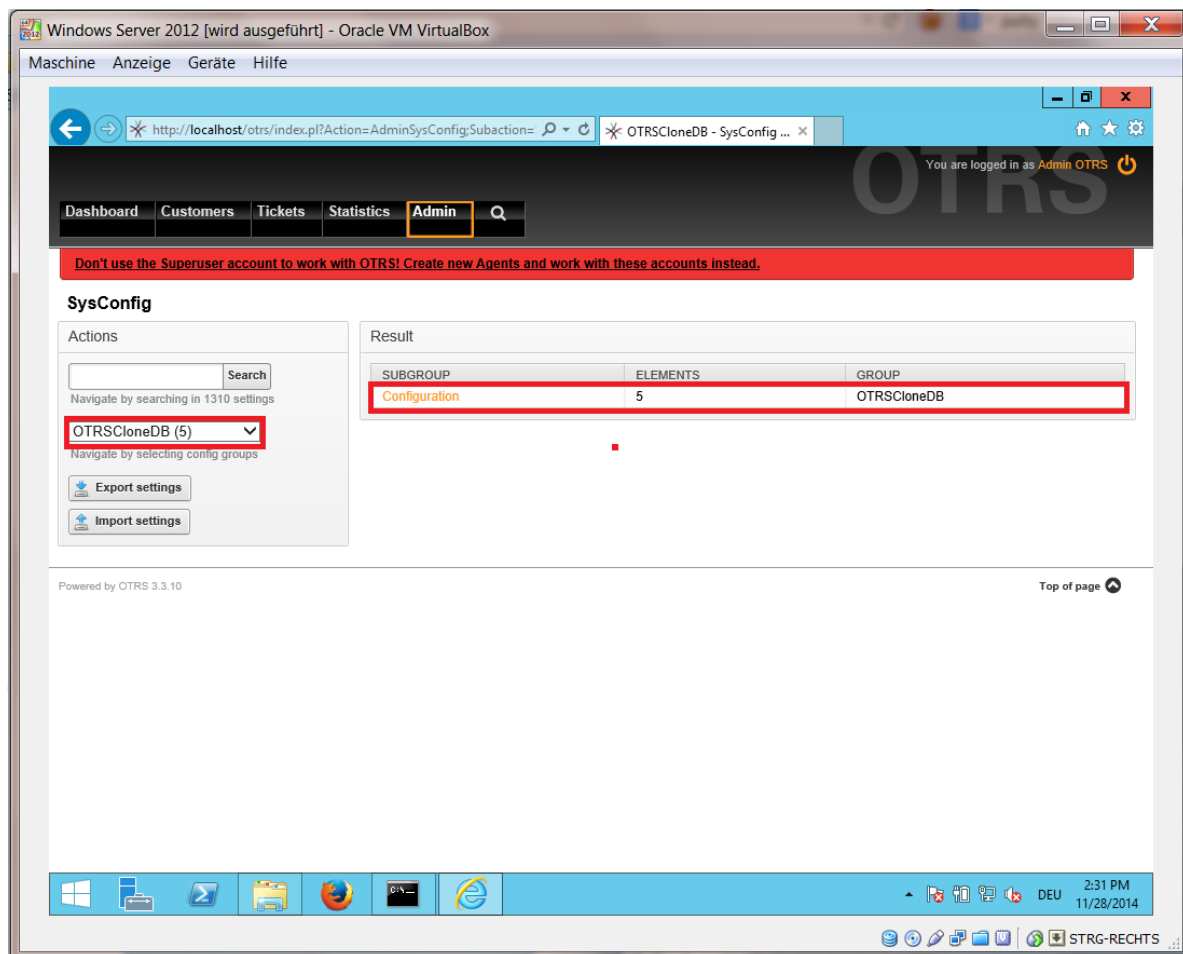
```
I88cm34A1B6xr13
```

Write the password down on a piece of paper.

4.1.7. Klonen Sie Ihre Datenbank in das Zielsystem

Switch back to your Windows based installation and open the SysConfig admin menu. Please select the group "OTRSCloneDB" and the subgroup "Configuration":

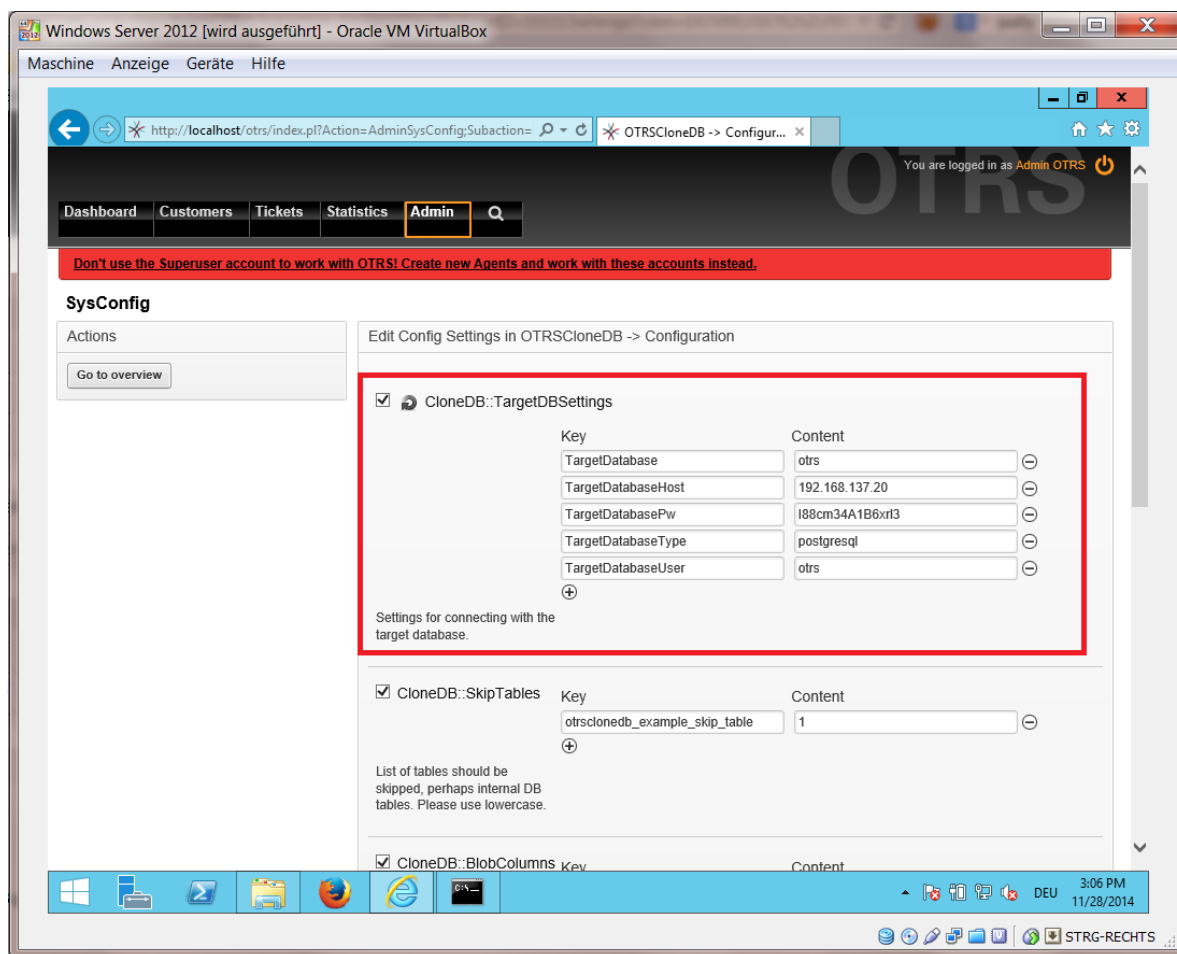
Abbildung 2.13. Konfiguration der OTRSCloneDB Sysconfig 1 - screenshot



Sie müssen in der SysConfig die Option OTRSCloneDB::TargetSettings mit den folgenden Werten konfigurieren:

```
TargetDatabaseHost => 192.168.137.20 (Here you need to enter the ip address of your target
system)
TargetDatabase => otrs
TargetDatabaseUser => otrs
TargetDatabasePw => I88cm34A1B6xrl3 (Here you need to set the password of your target
system)
TargetDatabaseType => postgresql
```

Abbildung 2.14. Konfiguration der OTRSCloneDB Sysconfig 2 - screenshot



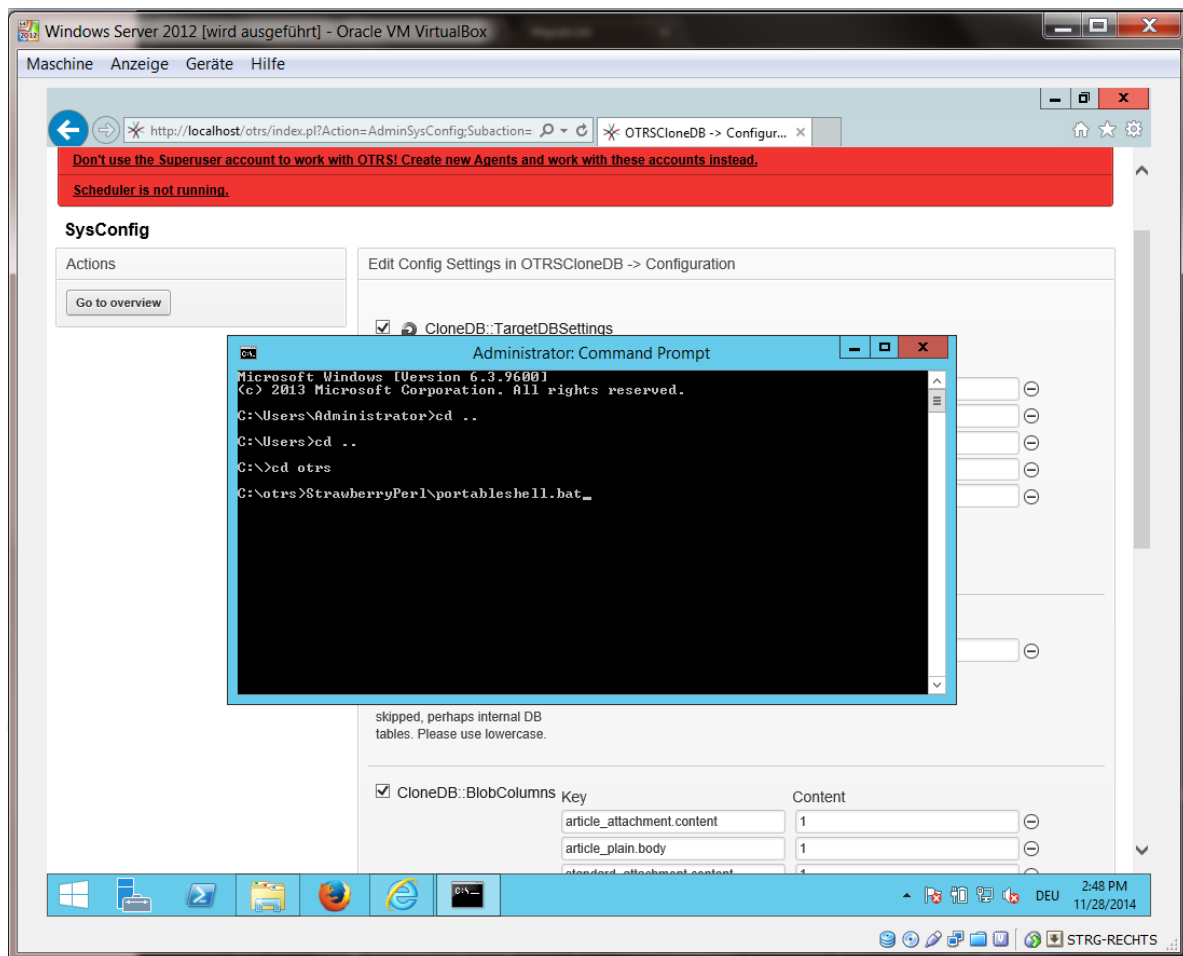
To run the OTRSCloneDB script we need to switch to the command prompt of our Windows based OTRS and to change into the base directory of our OTRS installation:

```
Shell> cd "C:\otrs"
```

If you are using StrawberryPerl, then you maybe need to activate your shell for Perl:

```
Shell> StrawberryPerl\portableshell.bat
```

Abbildung 2.15. Ausführen des OTRSCloneDB-Skript 1 - screenshot



The OTRSCloneDB script is located in the bin directory of the OTRS directory.

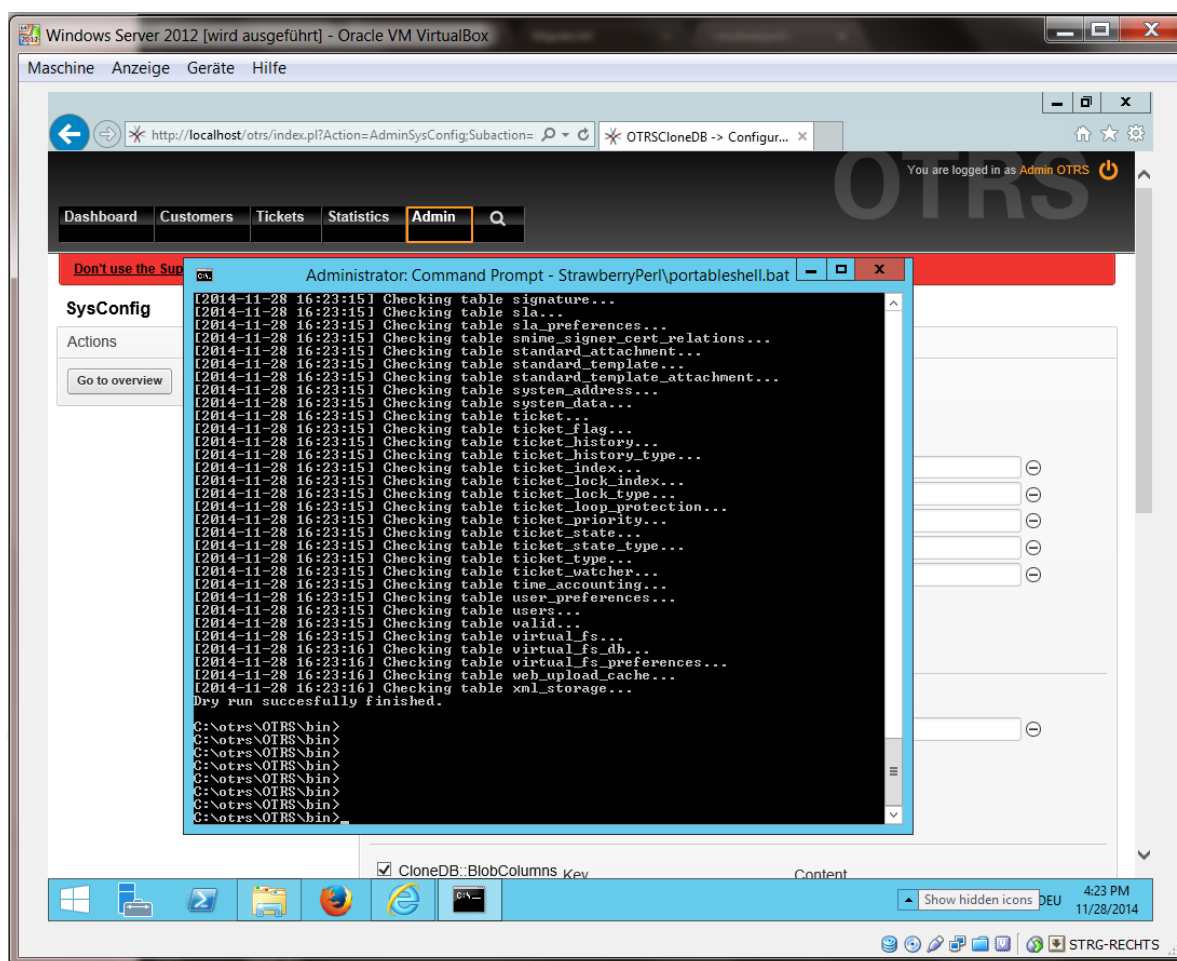
```
Shell> cd "OTRS\bin"
```

Führen Sie das OTRSCloneDB-Skript aus:

```
Shell> perl otrs.OTRSCloneDB.pl
```

Nun sollten Sie einige Informationen und Parameter des Skriptes sehen.

Abbildung 2.17. Run OTRSCloneDB script 3 - screenshot



Starten Sie das Klonen der Datenbank:

```
Shell> perl otrs.OTRSCloneDB.pl -r
```

Ein Beispiel für eine erfolgreiche Migration sieht wie folgt aus:

```
Generating DDL for OTRS.
Generating DDL for package OTRSCloneDB.
Creating structures in target database (phase 1/2)
...
...
Creating structures in target database (phase 2/2)
...
done.
```

4.1.8. Copy the following files from your Windows based system to the target system

You need to copy some files from your Windows based system to the target system. You can do this for example with a free tool like "WinSCP" (just search the internet for "WinSCP"). Copy the following files from your Windows based system:

```
C:\otrs\OTRS\Kernel\Config\GenericAgent.pm
```

```
C:\otrs\OTRS\Kernel\Config\Files\ZZZAuto.pm  
C:\otrs\OTRS\var\article\  
C:\otrs\OTRS\var\log\TicketCounter.log
```

zu Ihrem Zielsystem:

```
/opt/otrs/Kernel/Config/GenericAgent.pm  
/opt/otrs/Kernel/Config/Files/ZZZAuto.pm  
/opt/otrs/var/article/*  
/opt/otrs/var/log/TicketCounter.log
```

Open the file `/opt/otrs/Kernel/Config/Files/ZZZAuto.pm` on the target system and replace all paths like `"C:/otrs/OTRS/"` with `"/opt/otrs/"`!

If you have manually changes in your `Kernel/Config.pm` then please copy these changes to the target system's `Kernel/Config.pm`. Don't copy it 1:1 because you will now have different database settings and the file paths on the target system are different from Windows!

4.1.9. Alle Pakete erneut installieren

Reinstall all packages with the package manager to get all custom files back.

```
shell> bin/otrs.SetPermissions.pl --otrs-user=otrs --otrs-group=otrs --web-user=www-data --  
web-group=www-data /opt/otrs  
shell> su - otrs  
shell> cd /opt/otrs  
shell> perl bin/otrs.Console.pl Maint::Cache::Delete  
shell> perl bin/otrs.Console.pl Maint::Loader::CacheCleanup  
shell> perl bin/otrs.Console.pl Admin::Package::ReinstallAll  
shell> exit
```

Setzen Sie alle Berechtigungen in Ihrem OTRS-System erneut:

```
shell> bin/otrs.SetPermissions.pl --otrs-user=otrs --otrs-group=otrs --web-user=www-data --  
web-group=www-data /opt/otrs
```

4.1.10. Disable remote access for the PostgreSQL database of your target system

Undo all steps you did to enable the remote access for the PostgreSQL database to your target system.

Wechseln Sie in das PostgreSQL-Verzeichnis:

```
Shell> cd /etc/postgresql/9.4/main  
Shell> vi postgresql.conf
```

Entfernen Sie die folgenden Zeilen am Ende der Datei:

```
listen_addresses = '*'
```

Speichern Sie die Datei.


```
Shell> vi pg_hba.conf
```

Entfernen Sie die folgenden Zeilen am Ende der Datei:

```
host all all 0.0.0.0/0 md5
```

Speichern Sie die Datei.

Starten Sie Ihren postgresql-Server neu

```
Shell> service postgresql restart
```

4.1.11. OTRS-Dienste starten

Starten Sie Dienste auf Ihrem Zielsystem:

```
shell> service cron start  
shell> service apache2 start  
shell> su - otrs  
shell> cd /opt/otrs/  
shell> bin/Cron.sh start  
shell> bin/otrs.Daemon.pl -a start
```

Now you should be able to open the OTRS of your target system in the browser with the imported data of your Windows based system.

5. Upgrading OTRS from 4 to 5

These instructions are for people upgrading OTRS from 4 to 5 or from a 5 to a later patch-level release 5 and applies both for RPM and source code (tarball) upgrades.

If you are running a lower version of OTRS you have to follow the upgrade path to 4 first (1.1->1.2->1.3->2.0->2.1->2.2->2.3->2.4->3.0->3.1->3.2->3.3->4)! You need to perform a full upgrade to every version in between, including database changes and the upgrading Perl script.

Bitte beachten Sie, falls Sie von OTRS 2.2 oder früher upgraden, müssen Sie einen extra Schritt durchführen; bitte lesen Sie dazu 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 5 patchlevel 2 to version 5 patchlevel 6. If you need to do such a "patch level upgrade", you should skip steps 6, 10, 11, 14, 17 and 18.

Es wird empfohlen zuerst ein Test-Update auf einem separaten Testsystem durchzuführen

5.1. Schritt 1: Stoppen Sie alle relevanten Dienste.

Please make sure there are no more running services or cron jobs that try to access OTRS. This will depend on your service configuration, here is an example:

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

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

Stop OTRS cron jobs and the scheduler or daemon (in this order) depending on the OTRS version you are updating from:

```
shell> cd /opt/otrs/  
shell> bin/Cron.sh stop  
shell> bin/otrs.Scheduler.pl -a stop
```

or

```
shell> cd /opt/otrs/  
shell> bin/Cron.sh stop  
shell> bin/otrs.Daemon.pl stop
```

5.2. Schritt 2: Sichern Sie alles unter /opt/otrs/

- Kernel/Config.pm
- Kernel/Config/GenericAgent.pm (only for reference, this file is not needed any more)
- Kernel/Config/Files/ZZZAuto.pm
- var/*
- ebenfalls die Datenbank

5.3. Schritt 3: Stellen Sie sicher, dass Sie alles gesichert haben ;-)

5.4. Schritt 4: Installieren Sie das neue Release (tar oder RPM)

5.4.1. Schritt 4.1: Mit dem Tarball:

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

5.4.1.1. Wiederherstellen alter Konfiguration Dateien

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

5.4.1.2. Wiederherstellung TicketCounter.log

Damit OTRS mit der richtigen Ticketnummer weiter arbeitet, stellen Sie die Datei TicketCounter.log in \$OTRS_HOME/var/log/ wieder her. Das ist vorallem dann wichtig, wenn Sie inkrementelle Ticketnummern verwenden.

5.4.1.3. Wiederherstellung article data

If you configured OTRS to store article data in the filesystem you have to restore the article folder to /opt/otrs/var/ or the folder specified in the SysConfig.

5.4.1.4. Setzen Sie die Datei-Berechtigungen.

Please execute

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

with the permissions needed for your system setup. For example:

- Web server which runs as the OTRS user:

```
shell> bin/otrs.SetPermissions.pl --web-group=otrs
```

- Webserver with wwwrun user (e. g. SUSE):

```
shell> bin/otrs.SetPermissions.pl --web-group=wwwrun
```

- Webserver with apache user (e. g. Red Hat, CentOS):

```
shell> bin/otrs.SetPermissions.pl --web-group=apache
```

- Webserver with www-data user (e. g. Debian, Ubuntu):

```
shell> bin/otrs.SetPermissions.pl --web-group=www-data
```

5.4.2. Schritt 4.2: Mit RPM

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

In this case the RPM update automatically restores the old configuration files and sets file permissions.

5.5. Schritt 5: Überprüfung benötigter Perl-Module

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

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

5.6. Schritt 6: Anwenden der Datenbankänderungen

5.6.1. Schritt 6.1: Datenbankschema-Update

5.6.1.1. MySQL:

Anmerkung

Note: new tables created in the MySQL UPGRADING process will be created with the default table storage engine set in your MySQL server. In MySQL 5.5 the new default type is InnoDB. If existing tables, e.g. "users", have the table storage engine e.g. MyISAM, then an error will be displayed when creating the foreign key constraints. In this case we recommend to switch all tables to InnoDB with the console command **bin/otrs.Console.pl Maint::Database::MySQL::InnoDBMigration**.

Any problems with regards to the storage engine will be reported by the bin/otrs.Console.pl Maint::Database::Check command, so please run it to check for possible issues.

```
shell> cd /opt/otrs/  
shell> cat scripts/DBUpdate-to-5.mysql.sql | mysql -p -f -u root otrs  
shell> bin/otrs.Console.pl Maint::Database::Check
```

5.6.1.2. PostgreSQL:

```
shell> cd /opt/otrs/  
shell> cat scripts/DBUpdate-to-5.postgresql.sql | psql --set ON_ERROR_STOP=on --single-transaction otrs otrs
```

5.6.2. Schritt 6.2: Datenbank-Migrationskript

Run the migration script (as user otrs, NOT as root):

```
shell> scripts/DBUpdate-to-6.pl
```

The script will ask you to set a time zone for OTRS. It is very important that you set the correct time zone (and keep it), otherwise date and time of data added after the upgrade (tickets, articles, etc.) will be stored with a different time zone than your existing data, leading to inconsistent data. The script will suggest possible time zones based on your previous configuration. In case you are not sure or made a mistake, you can change the OTRS time zone after the upgrade via SysConfig setting OTRSTimeZone.

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

5.7. Step 7: Refresh the configuration cache and delete caches

Please run (as user otrs, *not* as root):

```
shell> cd /opt/otrs/  
shell> bin/otrs.Console.pl Maint::Config::Rebuild
```

```
shell> bin/otrs.Console.pl Maint::Cache::Delete
```

5.8. Step 8: Restart your services

e. g. (depends on used services):

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

Now you can log into your system.

5.9. Step 9: Check installed packages

Anmerkung

The OTRS packages of 4 are NOT compatible with OTRS 5, so you have to perform a package upgrade!

Die folgenden Pakete werden nach einem Upgrade automatisch deinstalliert (wenn sie zuvor installiert waren):

- OTRSGenericInterfaceMappingXSLT

5.10. Step 10: Configure NodeIDs (only for multi-frontend clustered setups)

Anmerkung

This step is only needed if you have a clustered setup with several frontend machines.

From OTRS 5 on, every frontend server needs to have its own unique NodeID. This defaults to 1 and thus does not need to be configured for single-frontend setups. If you have more than one machine, each machine needs to have this value set to a unique value between 1 and 999. This configuration needs to be done in the file `Kernel/Config.pm`:

```
$Self->{'NodeID'} = '2'; # assign a unique value for every frontend server
```

5.11. Step 11: Check follow-up detection configuration

The follow-up detection settings were reorganized. Now OTRS searches by default in email subject and references to detect follow-ups. Please check in `AdminSysConfig Ticket -> Core::PostMaster` if you need to make any changes to the follow-up detection configuration (for example to search in body, attachments or raw email content).

5.12. Step 12: Start the OTRS Daemon

The new OTRS daemon is responsible for handling any asynchronous and recurring tasks in OTRS. What has been in cron file definitions previously is now handled by the OTRS daemon, which is now required to operate OTRS. The daemon also handles all GenericAgent jobs and must be started from the `otrs` user.

```
shell> /opt/otrs/bin/otrs.Daemon.pl start
```

5.13. Step 13: Update and activate cron jobs

There are two default OTRS cron files in `/opt/otrs/var/cron/*.dist`, and their purpose is to make sure that the OTRS Daemon is running. They need to be activated by copying them without the ".dist" filename extension.

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

To schedule these cron jobs on your system, you can use the script `Cron.sh` with the `otrs` user.

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

Please note that if you had any custom cron jobs, you should consider moving them to `SysConfig (Daemon -> Daemon::SchedulerCronTaskManager::Task)` to have them executed by the OTRS daemon as well. You might also need to adapt your custom scripts, because now most OTRS commands are managed by `bin/otrs.Console.pl` instead of single scripts.

5.14. Step 14: Review your ticket notifications

With OTRS 5 ticket notifications are configured differently than in previous versions. They are now part of the "Event Notifications" that were previously available as well (now called just "Ticket Notifications"). Your existing ticket notifications have been migrated into the list of ticket notifications, but deactivated. You have also received the new default ticket notifications, active by default.

In case you did not modify the old ticket notifications you don't have to do anything. If you have made local modifications or translations, you have two choices: a) you can migrate your modifications to the new ticket notifications (recommended) or b) you can deactivate the new notifications and activate the old notifications again.

Please also review the escalation notification recipient settings and see if they match with your expectations (agents who have the ticket in one of their custom queues vs. all agents with read permissions). This was configured via a setting in the file `Kernel/Config/GenericAgent.pm` (now obsolete) previously and can now be controlled via the GUI for each notification separately.

5.15. Step 15: Update system registration (optional)

If the system is already registered with OTRS Group, it is strongly recommended to update the registration information at this time. This will update the registered version of the system (among other changes) in the OTRS Group records, in order to get much accurate information from the cloud services.

If you don't update the registration information manually, it will be done automatically on a regular basis, but this could happen some hours or days after. During this period

it might be possible to get wrong information from cloud services like **OTRS Business Solution™** updates.

```
shell> cd /opt/otrs/  
shell> bin/otrs.Console.pl Maint::Registration::UpdateSend --force  
shell> bin/otrs.Console.pl Maint::Cache::Delete
```

5.16. Step 16: Migrate custom Perl based GenericAgent jobs (optional)

This is only relevant if you had any custom GenericAgent jobs in Perl OTRS 4 configuration files such as Kernel/Config/GenericAgent.pm that cannot be handled with the regular ticket notifications. Such jobs now need to be registered as OTRS Daemon cron tasks in the SysConfig (Daemon -> Daemon::SchedulerCronTaskManager::Task), in order to be executed on a regular basis.

There are 5 settings in the SysConfig prepared for this purpose (Daemon::SchedulerCronTaskManager::Task###GenericAgentFile1 to Daemon::SchedulerCronTaskManager::Task###GenericAgentFile5). If more settings are needed they can be added in a custom SysConfig setting file.

Please replace "<ModuleName>" with the file that contains the custom GenericAgent jobs configuration, e.g. Kernel::Config::GenericAgent for the file: Kernel/Config/GenericAgent.pm, set the schedule to match the periodicity as it was executed before, mark the setting as active and save the changes.

Anmerkung

To get more information about other parameters that can be used while running the GenericAgent jobs, please execute:

```
shell> bin/otrs.Console.pl Maint::GenericAgent::Run --help
```

5.17. Step 17: Setup bash autocompletion (optional)

All regular OTRS command line operations now happen via the OTRS Console interface bin/otrs.Console.pl. This provides an auto completion for the bash shell which makes finding the right command and options much easier.

You can activate the bash autocompletion by installing the package bash-completion. It will automatically detect and load the file /opt/otrs/.bash_completion for the otrs user.

After restarting your shell, you can just type bin/otrs.Console.pl followed by TAB, and it will list all available commands. If you type a few characters of the command name, TAB will show all matching commands. After typing a complete command, all possible options and arguments will be shown by pressing TAB.

5.18. Step 18: Review ticket action screen configurations (optional)

Some ticket action screens like "Note" had default subjects in OTRS 4 (configurable via SysConfig, Ticket::Frontend::AgentTicketNote###Subject in this case). These

default subjects have been removed to reduce the amount of potentially redundant/meaningless data. You might want to re-add them if they are mandatory for you.

5.19. Step 19: Well done!

6. Additional Applications

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.

6.1. 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 3. First Steps

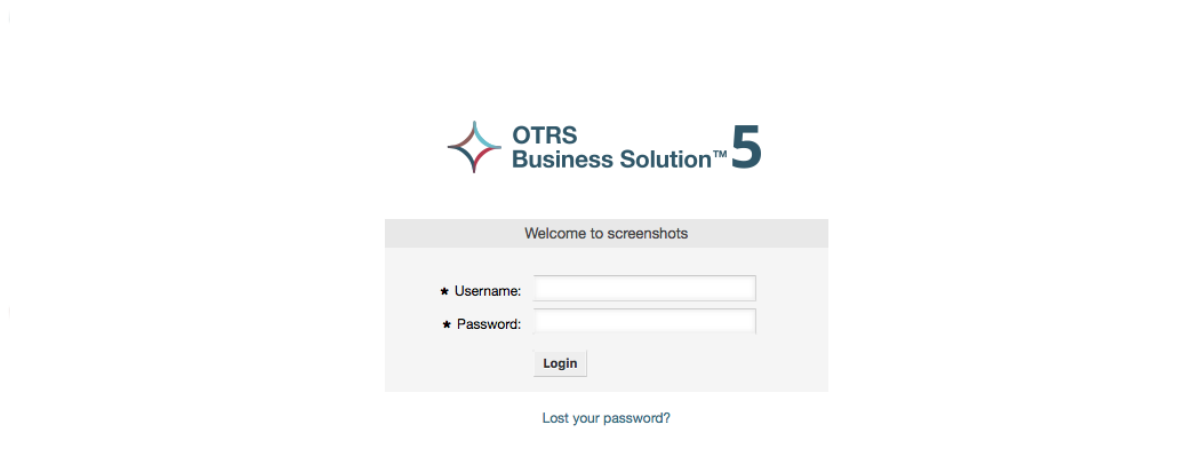
Dieser Abschnitt soll einen ersten Überblick über die Funktionsweise von OTRS und den Aufbau der Weboberfläche des Systems geben. Es werden die Unterschiede zwischen den Mitarbeitern 'agents', den Kunden 'Customer' und Administratoren 'administrators' erklärt. Anschließend wird die erste Anmeldung als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

1. Agent Web Interface

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

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

Abbildung 3.1. Login screen of the agent interface



2. Customer Web Interface

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

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

Abbildung 3.2. Login screen of the customer interface

Example Company

Login

Forgot password?

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


Switch to mobile mode

3. Public Web Interface

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

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

Abbildung 3.3. Öffentliche Weboberfläche

Example Company 

FAQ Explorer

Subcategories

NAME	COMMENT	SUBCATEGORIES	FAQ ARTICLES
Bugzilla	Reporting bugs	0	1
Feature AddOns	FAQs	0	1
OTRS Framework	Questions and answers about OTRS	0	1

FAQ Articles

FAQ#	TITLE	CATEGORY	LANGUAGE
No FAQ articles found.			

Search

Wildcards are allowed.

Q Advanced Search

Latest created FAQ articles RSS

Advanced Escalations

Feature AddOns - en - public (all) - 12/03/2014 00:54

Framework bug

Bugzilla - en - public (all) - 12/03/2014 00:53

Test Uno

OTRS Framework - en - public (all) - 12/03/2014 00:51

Latest updated FAQ articles RSS

Advanced Escalations

Feature AddOns - en - public (all) - 12/03/2014 00:54

Framework bug

Bugzilla - en - public (all) - 12/03/2014 00:53

Test Uno

OTRS Framework - en - public (all) - 12/03/2014 00:51

Top 10 FAQ articles RSS

4. First Login

Access the login screen as described in the section Agent web interface . Enter a user name and password. Since the system has just been installed and no users have yet been

created, login as OTRS administrator first, using 'root@localhost' for username and 'root' for password.

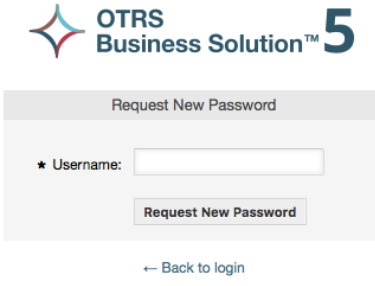
Warnung

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.

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

Abbildung 3.4. Neues Passwort beantragen

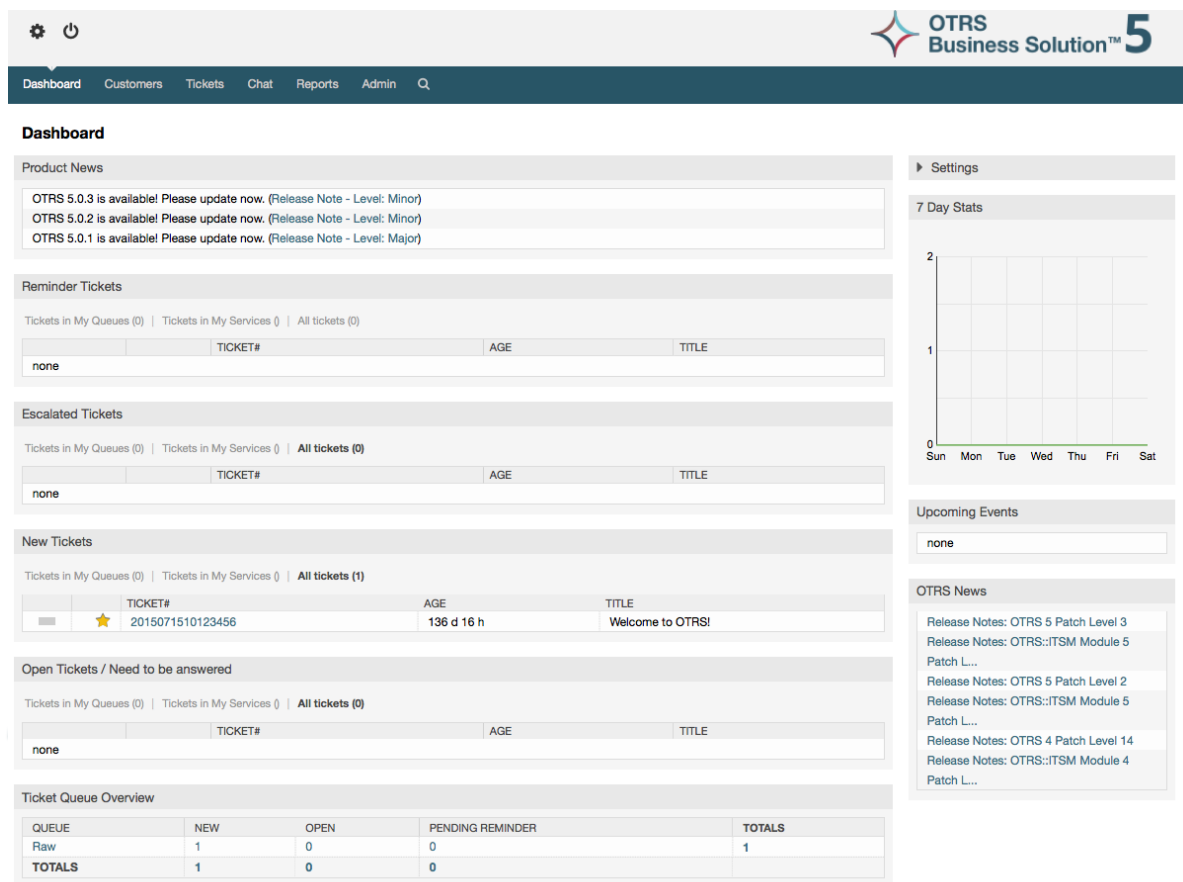


The screenshot shows the OTRS Business Solution 5 interface. At the top, there is a logo for OTRS Business Solution 5. Below the logo is a form titled "Request New Password". The form contains a single input field labeled "★ Username:" and a button labeled "Request New Password". Below the button is a link that says "← Back to login".

5. The Web Interface - an Overview

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

Abbildung 3.5. Dashboard of the agent interface



Product News

OTRS 5.0.3 is available! Please update now. (Release Note - Level: Minor)
 OTRS 5.0.2 is available! Please update now. (Release Note - Level: Minor)
 OTRS 5.0.1 is available! Please update now. (Release Note - Level: Major)

Reminder Tickets

Tickets in My Queues (0) | Tickets in My Services (0) | All tickets (0)

	TICKET#	AGE	TITLE
none			

Escalated Tickets

Tickets in My Queues (0) | Tickets in My Services (0) | All tickets (0)

	TICKET#	AGE	TITLE
none			

New Tickets

Tickets in My Queues (0) | Tickets in My Services (0) | All tickets (1)

	TICKET#	AGE	TITLE
★	2015071510123456	136 d 16 h	Welcome to OTRS!

Open Tickets / Need to be answered

Tickets in My Queues (0) | Tickets in My Services (0) | All tickets (0)

	TICKET#	AGE	TITLE
none			

Ticket Queue Overview

QUEUE	NEW	OPEN	PENDING REMINDER	TOTALS
Raw	1	0	0	1
TOTALS	1	0	0	

Settings

7 Day Stats

0 1 2

Sun Mon Tue Wed Thu Fri Sat

Upcoming Events

none

OTRS News

- Release Notes: OTRS 5 Patch Level 3
- Release Notes: OTRS::ITSM Module 5 Patch L...
- Release Notes: OTRS 5 Patch Level 2
- Release Notes: OTRS::ITSM Module 5 Patch L...
- Release Notes: OTRS 4 Patch Level 14
- Release Notes: OTRS::ITSM Module 4 Patch L...

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

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

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

In the area below the navigation bar, different system messages can be shown. If you are logged in as the default OTRS administrator user, you get a red message warning you not to work using this system account.

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

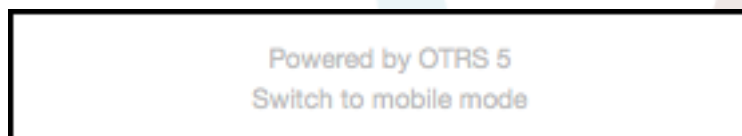
These boxes contains the main part of each screen, usually they are displayed in one or several columns, each box can store relevant information about the current screen like for

example instructions, advises, overviews, etc. Also is displayed the form or tool necessary for performing the action associated to each screen, like for example, add, update or delete records, check the log, change configuration settings, etc.

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

Normalerweise sehen die "icon row", die "navigation bar" und die Fußzeile überall im gesamten Webinterface gleich aus.

Abbildung 3.6. Fußzeile



6. Die Übersichtsseite

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

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

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

Abbildung 3.7. Übersichtsseiten-Widgets

Dashboard

Product News

Can't connect to Product News server!

Reminder Tickets

Tickets in My Queues (0) | Tickets in My Services () | All tickets (0)

		TICKET#	AGE	TITLE
none				

Escalated Tickets

Tickets in My Queues (0) | Tickets in My Services () | All tickets (0)

		TICKET#	AGE	TITLE
none				

New Tickets

Tickets in My Queues (0) | Tickets in My Services () | All tickets (1)

		TICKET#	AGE	TITLE
■	★	2015071510123456	136 d 16 h	Welcome to OTRS!

Open Tickets / Need to be answered

Tickets in My Queues (0) | Tickets in My Services () | All tickets (0)

		TICKET#	AGE	TITLE
none				

Ticket Queue Overview

QUEUE	NEW	OPEN	PENDING REMINDER	TOTALS
Raw	1	0	0	1
TOTALS	1	0	0	

Linke Spalte der Übersichtsseiten-Widgets.

- Ticketlisten Widgets

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

- Meine gesperrten Tickets

Die Tickets, die der eingeloggte Agent gesperrt hat.

- Meine beobachteten Tickets

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

- Meine Verantwortlichkeiten

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

- Tickets in meinen Queues

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

- Ticket in MyServices

The tickets that are assigned to services where the agent define as "My Services" and are on queues with at least read-only permissions.

- Alle Tickets

Alle Tickets auf die der Agent Zugriff hat.

Das sind folgende Widgets:

- Erinnerungs-Tickets

Tickets die auf 'warten' gesetzt sind und deren Erinnerungszeit erreicht ist.

- Eskaliert

Tickets die eskaliert sind.

- Neue Tickets

Tickets die den Status "Neu" haben.

- Offene Tickets / Beantwortung erforderlich

Tickets die den Status "Offen" haben.

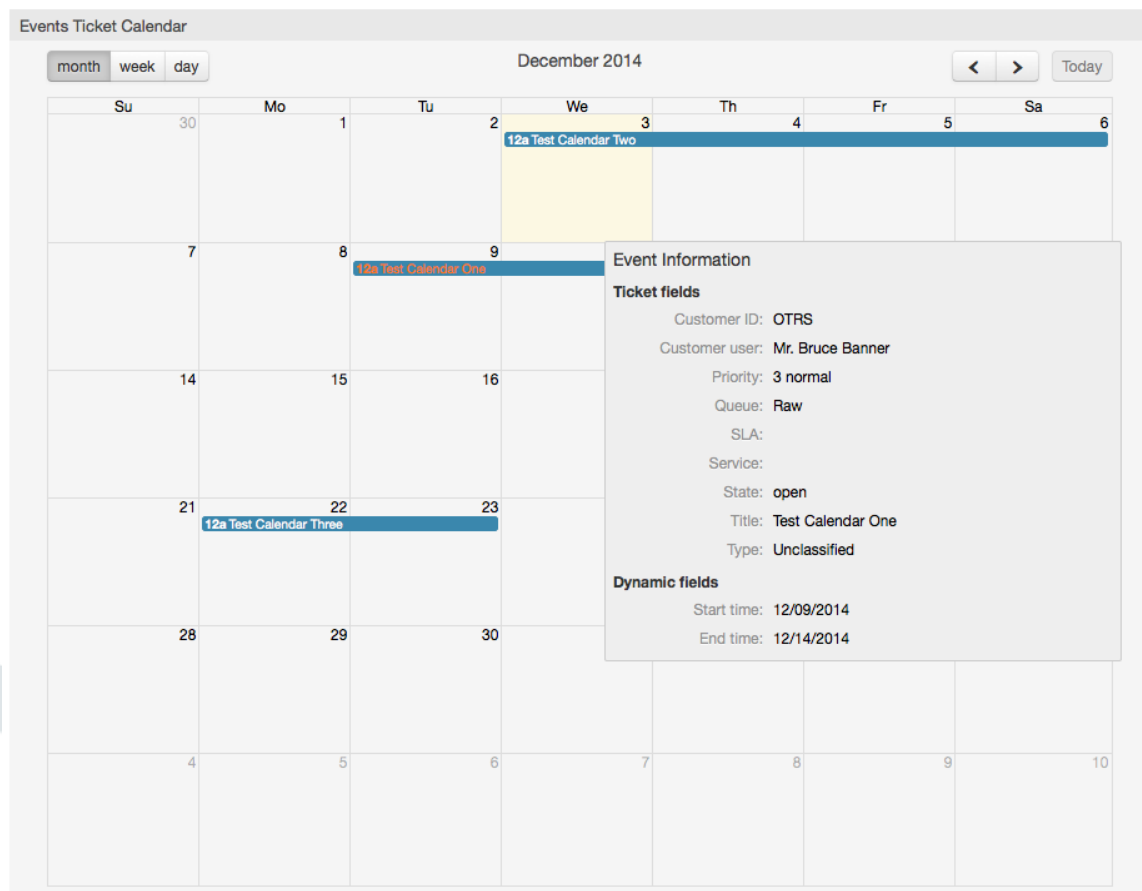
- Ticket-Ereigniskalender

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

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

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

Abbildung 3.8. Ticket Ereigniskalender Widget



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

- **Kalenderbreite**
Definiert die Kalenderbreite in Prozent. Standard ist 95%
- **DynamicFieldStartTime**
Definiert den Namen des dynamischen Feldes für die Startzeit.
- **DynamicFieldEndTime**
Definiert den Namen des dynamischen Feldes für die Endzeit.
- **Queues**
Only the tickets on the queues specified in this setting will be considered in the calendar view.
- **DynamicFieldsForEvents**
Definiert die dynamischen Felder die im Kalender-Overlay angezeigt werden.
- **TicketFieldsForEvents**
Definiert die Ticketattribute die im Kalender-Overlay angezeigt werden.

- Übersicht nach Queues

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

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

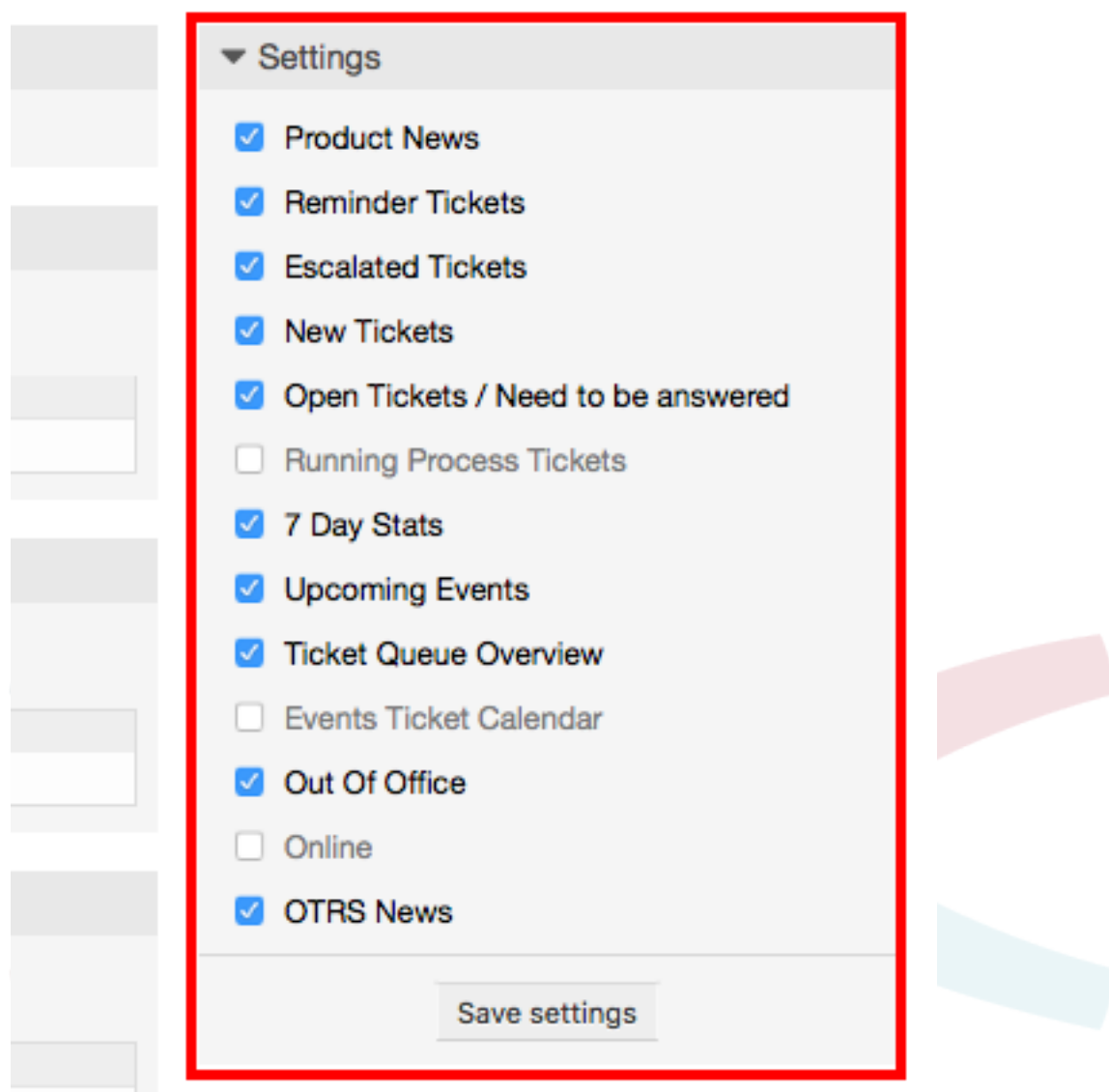
The queues and states that are presented can be changed via Sysconfig.

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

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



Abbildung 3.9. Dashboard Settings



Rechte Spalte der Übersichtsseiten-Widgets.

- 7 Tage-Statistik

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

- Anstehende Ereignisse

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

- OTRS Neuigkeiten

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

- Online

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

7. What is a 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 gespeichert 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 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.

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

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

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

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

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.

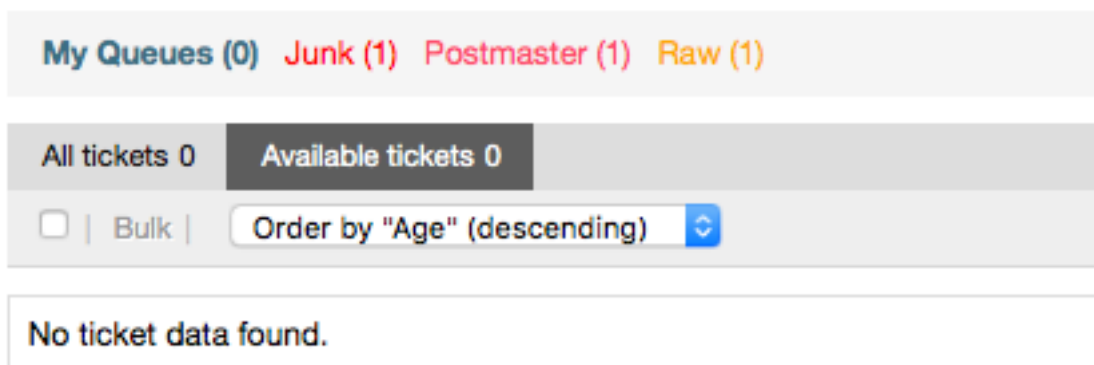
Je besser und strukturierter ein Support-system organisiert ist, desto weniger Zeit und letztlich auch finanzielle Mittel müssen dafür aufgebracht werden. Queues und Unter-Queues helfen bei der Strukturierung bzw. bei der Abbildung von Abläufen.

8. What is the Queue Overview?

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

Abbildung 3.10. Queue-Ansicht (Standard) für Agenten

QueueView: My Queues



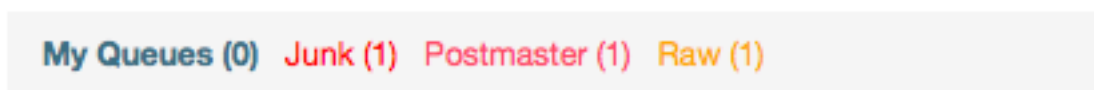
The queue overview offers a variety of options for daily work with OTRS. The first of these is the My Queue. In the Agent Preferences, or when administering agents, a set of queues can be defined for which the agent has been assigned to work within. All the tickets will appear in this default view, when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

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

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

Es gibt visuelle Alarme, um den Agent zu unterstützen

Abbildung 3.11. Visuelle Alarme der Queueansicht



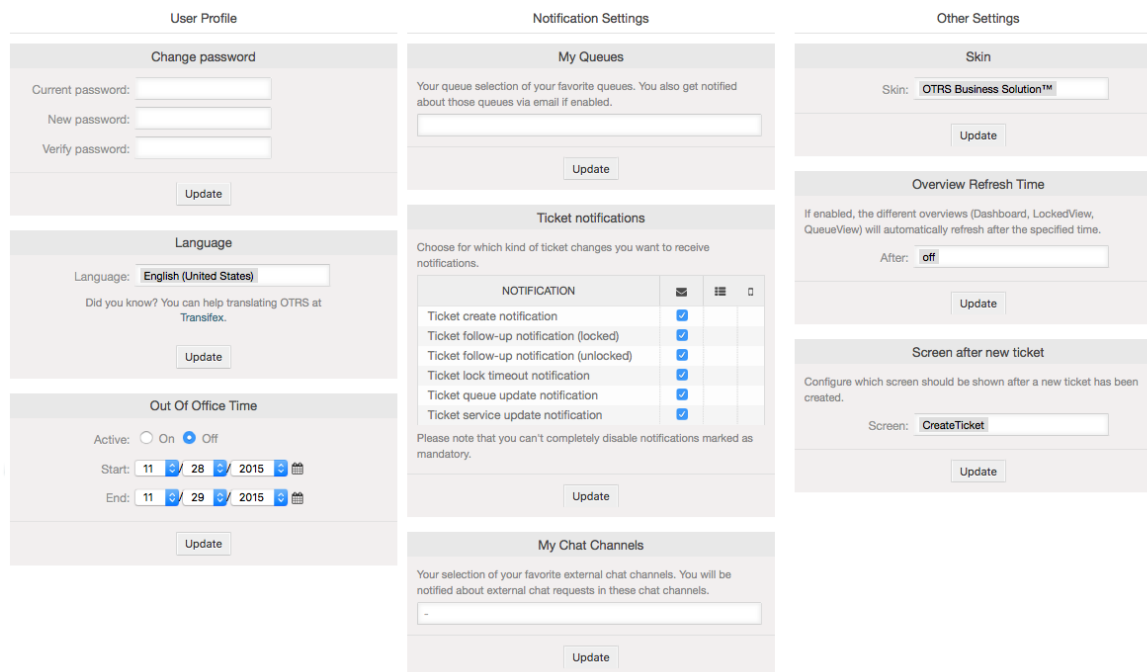
visuelle Alarme

- Highlight Age 1: Sets the age in minutes (first level) for highlighting queues that contain untouched tickets. Seen in above in the "Raw" queue.
- Highlight Age 2: Sets the age in minutes (second level) for highlighting queues that contain untouched tickets. Seen in above in the "Postmaster" queue.
- Blink: Activates a blinking mechanism of the queue that contains the oldest ticket. Not supported in all browsers. In that case it will appear red, as seen in above in the "Junk" queue.
- Bold: The current queue will be bolded, as seen above in the "My Queues".

9. User Preferences

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

Abbildung 3.12. Agent's personal preferences



The screenshot shows the 'Agent's personal preferences' configuration page, organized into three main sections:

- User Profile:**
 - Change password:** Fields for Current password, New password, and Verify password, with an 'Update' button.
 - Language:** A dropdown menu set to 'English (United States)', a note about Transifex, and an 'Update' button.
 - Out Of Office Time:** Radio buttons for 'Active' (Off), 'Start' and 'End' date/time pickers (e.g., 11/28/2015), and an 'Update' button.
- Notification Settings:**
 - My Queues:** A text input field for queue selection and an 'Update' button.
 - Ticket notifications:** A table to select notification types.

NOTIFICATION	✉	📧	🔊
Ticket create notification	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ticket follow-up notification (locked)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ticket follow-up notification (unlocked)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ticket lock timeout notification	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ticket queue update notification	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ticket service update notification	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 - My Chat Channels:** A text input field for chat channel selection and an 'Update' button.
- Other Settings:**
 - Skin:** A dropdown menu set to 'OTRS Business Solution™' and an 'Update' button.
 - Overview Refresh Time:** A text input field set to 'off' and an 'Update' button.
 - Screen after new ticket:** A dropdown menu set to 'CreateTicket' and an 'Update' button.

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

Benutzerprofil

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

Benachrichtigungseinstellungen

- Wählen Sie die Queues aus, die sie in "Meine Queues" überwachen wollen.
- Wählen Sie die Dienste aus, die Sie in "Meine Dienste" überwachen möchten.
- Configure which ticket notifications you want to receive (per transport method).

Andere Einstellungen

- Oberflächen-Skin auswählen.
- Set the refresh period for the overviews (Dashboard, Queue View, etc.).

- Bildschirm auswählen, der nach Erstellung eines Tickets angezeigt wird.

Abbildung 3.13. Customer's personal preferences

Example Company

Tickets

Interface language

Language

Number of displayed tickets

Tickets per page

Ticket overview

Refresh interval

Change password

Current password

New password

Verify password

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 4. Administration

1. The Administration Area of OTRS

1.1. Grundlagen

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

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

Abbildung 4.1. OTRS Administration Overview Screen

Agent Management Agents Create and manage agents. Agents <-> Groups Link agents to groups. Agents <-> Roles Link agents to roles.		Customer Management Customer User Create and manage customer users. Customer User <-> Groups Link customer user to groups.		Customer Management Customers Create and manage customers. Customer User <-> Services Link customer user to services.		Email Settings PostMaster Mail Accounts Manage POP3 or IMAP accounts to fetch email from. Email Addresses Set sender email addresses for this system. PGP Keys Manage PGP keys for email encryption.		Email Settings PostMaster Filters Filter incoming emails. S/MIME Certificates Manage S/MIME certificates for email encryption.			
Queue Settings Queues Create and manage queues. Templates <-> Queues Link templates to queues. Auto Responses <-> Queues Link queues to auto responses.		Queue Settings Templates Create and manage templates. Auto Responses Create and manage responses that are automatically sent. Attachments Create and manage attachments. Salutations Create and manage salutations.		Ticket Settings Ticket Notifications Create and manage ticket notifications. Access Control Lists (ACL) Configure and manage ACLs. Priorities Create and manage ticket priorities. Dynamic Fields Create and manage dynamic fields.		Ticket Settings Types Create and manage ticket types. States Create and manage ticket states. Services Create and manage services. Service Level Agreements Create and manage Service Level Agreements (SLAs).		System Administration GenericAgent Manage tasks triggered by event or time based execution. OTRS Business Solution™ Deploy and manage OTRS Business Solution™. Cloud Services Manage OTRS Group cloud services. Session Management Manage existing sessions. Performance Log View performance benchmark results. SQL Box Execute SQL statements. SysConfig Edit the system configuration settings. Package Manager Update and extend your system with software packages.		System Administration System Registration Manage system registration. Support Data Collector Manage support data. Admin Notification Send notifications to users. System Maintenance Schedule a maintenance period. System Log View system log messages. Process Management Configure Processes. Web Services Create and manage web services. Chat Channel Create and manage chat channels.	

1.2. Benutzer, Gruppen und Rollen

1.2.1. Benutzer

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

Anmerkung

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

Abbildung 4.2. Agenten-Verwaltung

Agent Management

Actions

Wildcards like "*" are allowed.

Hint

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

USERNAME	NAME	EMAIL	LAST LOGIN	VALIDITY	CHANGED	CREATED
carlos.garcia	Carlos Garcia	carlos.garcia@mycompany...	11/26/2015 12:10	valid	11/25/2015 13:25	11/25/2015 13:25
carlos.rodriiguez	Carlos Rodríguez	carlos.rodriiguez@mycomp...		valid	11/25/2015 13:25	11/25/2015 13:25
dennis.schmelter	Dennis Schmelter	dennis.schmelter@mycomp...		valid	11/25/2015 13:25	11/25/2015 13:25
dominik.klein	Dominik Klein	dominik.klein@mycompany...		valid	11/25/2015 13:25	11/25/2015 13:25
jan.steinweg	Jan Steinweg	jan.steinweg@mycompany.com		valid	11/25/2015 13:25	11/25/2015 13:25
jens.pfeifer	Jens Pfeifer	jens.pfeifer@mycompany.com		valid	11/25/2015 13:25	11/25/2015 13:25
manuel.hecht	Manuel Hecht	manuel.hecht@mycompany.com		valid	11/25/2015 13:25	11/25/2015 13:25
marc.bonsels	Marc Bonsels	marc.bonsels@mycompany.com		valid	11/25/2015 13:25	11/25/2015 13:25
marco.buchholz	Marco Buchholz	marco.buchholz@mycompan...		valid	11/25/2015 13:25	11/25/2015 13:25
martin.gruner	Martin Gruner	martin.gruner@mycompany...		valid	11/25/2015 13:25	11/25/2015 13:25
oliver.rottges	Oliver Rottges	oliver.rottges@mycompan...		valid	11/25/2015 13:25	11/25/2015 13:25
patrick.brischler	Patrick Brischler	patrick.brischler@mycom...		valid	11/25/2015 13:25	11/25/2015 13:25
root@localhost	Admin OTRS	root@localhost	11/26/2015 11:58	valid	11/25/2015 13:25	11/25/2015 13:25
udo.bretz	Udo Bretz	udo.bretz@mycompany.com		valid	11/25/2015 13:25	11/25/2015 13:25

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

Abbildung 4.3. Adding a new agent

Agent Management

Actions

Hint

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

Add Agent

Title:

★ **Firstname:**

★ **Lastname:**

★ **Username:**

Password:

Will be auto-generated if left empty.

★ **Email:**

Mobile:

Validity:

Google Authenticator:

Enter your shared secret to enable two factor authentication.

Language:

Language

Out Of Office Time: On Off

Start: / /

End: / /

Skin:

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.

1.2.2. Gruppen

Every agent's account should belong to at least one group or role. In a brand new installation, there are three pre-defined groups available, as shown in Table 4-1.

Tabelle 4.1. 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 Statistik-

Gruppe	Beschreibung
	modul 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".

You can access the group management page (see figure below) by clicking the *Groups* link in the admin area.

Abbildung 4.4. Gruppen-Verwaltung

Group Management

Actions

+ Add group

Hint

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

List

NAME	COMMENT	VALIDITY	CHANGED	CREATED
admin	Group of all administrators.	valid	11/19/2015 13:25	11/19/2015 13:25
stats	Group for statistics access.	valid	11/19/2015 13:25	11/19/2015 13:25
users	Group for default access.	valid	11/19/2015 13:25	11/19/2015 13:25

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.

To add an agent to a group, or to change the agents who belong to a group, you can use the link *Agents <-> Groups* from the Admin page (see figure below).

Abbildung 4.5. Agent <-> group management

Manage Agent-Group Relations

Actions

+ Add agent

+ Add group

Filter for Agents

Just start typing to filter...

Filter for Groups

Just start typing to filter...

Overview

AGENTS

- carlos.garcia (Carlos Garcia)
- carlos.rodriguez (Carlos Rodríguez)
- dennis.schmelter (Dennis Schmelter)
- dominik.klein (Dominik Klein)
- jan.steinweg (Jan Steinweg)
- jens.pfeifer (Jens Pfeifer)
- manuel.hecht (Manuel Hecht)
- marc.bonsels (Marc Bonsels)
- marco.buchholz (Marco Buchholz)
- martin.gruner (Martin Gruner)
- oliver.rottges (Oliver Rottges)
- patrick.brischler (Patrick Brischler)
- udo.bretz (Udo Bretz)

GROUPS

- admin
- stats
- users

An overview of all groups and agents in the system is displayed on this page. You can also use the available filters to find a specific entity. If you want to change the groups that an

agent is a member of, just click on the agent's name (see figure below). To change the agents associated with a group, just click on the group you want to edit (see figure below).

Abbildung 4.6. Change the groups an agent belongs to

Manage Agent-Group Relations

Actions: [Go to overview](#)

Filter:

Change Group Relations for Agent Dominik Klein (dominik.klein)

GROUP	<input type="checkbox"/> CHAT_OBSERVER	<input type="checkbox"/> CHAT_PARTICIPANT	<input type="checkbox"/> CHAT_OWNER	<input type="checkbox"/> RO	<input type="checkbox"/> MOVE_INTO	<input type="checkbox"/> CREATE	<input type="checkbox"/> NOTE	<input type="checkbox"/> OWNER	<input type="checkbox"/> PRIORITY	<input type="checkbox"/> RW
admin	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
stats	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

or

Reference

- ro**
Read only access to the ticket in this group/queue.
- move_into**
Permissions to move tickets into this group/queue.
- create**
Permissions to create tickets in this group/queue.
- note**
Permissions to add notes to tickets in this group/queue.
- owner**
Permissions to change the owner of tickets in this group/queue.
- priority**
Permissions to change the ticket priority in this group/queue.
- rw**
Full read and write access to the tickets in this group/queue.

Abbildung 4.7. Change the agents that belong to a specific group

Manage Agent-Group Relations

Actions: [Go to overview](#)

Filter:

Change Agent Relations for Group users

AGENT	<input type="checkbox"/> CHAT_OBSERVER	<input type="checkbox"/> CHAT_PARTICIPANT	<input type="checkbox"/> CHAT_OWNER	<input type="checkbox"/> RO	<input type="checkbox"/> MOVE_INTO	<input type="checkbox"/> CREATE	<input type="checkbox"/> NOTE	<input type="checkbox"/> OWNER	<input type="checkbox"/> PRIORITY	<input type="checkbox"/> RW
carlos.garcia (Carlos Garcia)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
carlos.rodriguez (Carlos Rodriguez)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
dennis.schmelter (Dennis Schmelter)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
dominik.klein (Dominik Klein)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
jan.steinweg (Jan Steinweg)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
jens.pfeifer (Jens Pfeifer)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
manuel.hecht (Manuel Hecht)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
marc.bonsels (Marc Bonsels)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
marco.buchholz (Marco Buchholz)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
martin.gruner (Martin Gruner)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
oliver.rottges (Oliver Rottges)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
patrick.brischler (Patrick Brischler)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
root@localhost (Admin OTRS)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
udo.bretz (Udo Bretz)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

or

Each group has a set of rights associated with it, and each group member (agent) may have some combination of these rights for themselves. A list of the permissions / rights is shown in Table 4-2.

Tabelle 4.2. Rights associated with OTRS groups

Berechtigung	Beschreibung
chat_observer	Agents may take part silently in a chat (available in OTRS Business Solution™).
chat_participant	Agents may normally participate in a chat (available in OTRS Business Solution™).
chat_owner	Agents have full rights for a chat and can accept chat requests (available in OTRS Business Solution™).

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

Anmerkung

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

Nicht alle verfügbaren Berechtigungen werden standardmäßig angezeigt. Diese zusätzlichen Berechtigungen können hinzugefügt werden.

Tabelle 4.3. Weitere Berechtigungsgruppen

Berechtigung	Beschreibung
stats	Erlaubt den Zugriff auf die Statistikseite
Umleiten	Das Recht eine E-Mail umzuleiten (mit Umleiten-Knopf im TicketZoom)
Verfassen	Das Recht eine Antwort zu einem Ticket zu verfassen.
Kunde	Das Recht den Kunden eines Tickets zu ändern.
Weiterleiten	Das Recht eine Nachricht weiterzuleiten (mit dem Weiterleiten-Knopf)
Warten	Das Recht ein Ticket auf "warten" zu setzen.
Telefon	Das Recht einen Telefonanruf zu einem Ticket hinzuzufügen.
Verantwortlicher	Das Recht den Verantwortlichen Agenten eines Tickets zu ändern.

Anmerkung

Diese Berechtigungen können durch ändern von `System::Permission` hinzugefügt werden

1.2.3. Rollen

Rollen sind ein sehr 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

Roles are really useful when dealing with complex organizations and when maintaining larger OTRS installations. Proper care is advised though. Mixing Agent to Group with Agent to Role mappings can make for a complex access control scheme, that is difficult to understand and maintain. If you wish to use only roles and disable the Agents <-> Groups option in the Admin area, you can do so by modifying the Frontend::Module###AdminUserGroup in the SysConfig. Be aware that this won't remove already existing Agents to Group assignments!

You can access the role management section (see figure below) by clicking the *Roles* link on the Admin page.

Abbildung 4.8. Rollenverwaltung

Role Management

Actions

+ Add role

Hint

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

List

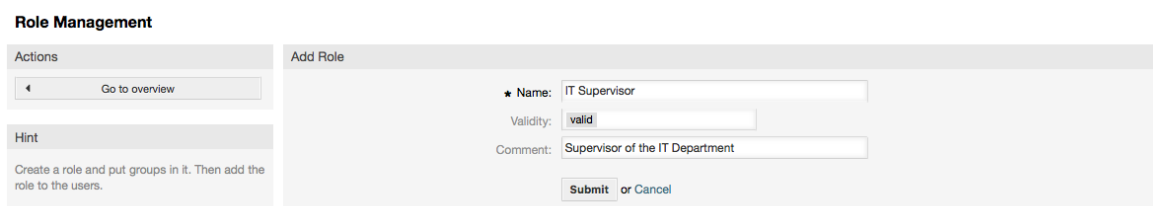
NAME	COMMENT	VALIDITY	CHANGED	CREATED
Development		valid	12/03/2015 19:34	12/03/2015 19:34
IT Supervisor	Supervisor of the IT De...	valid	12/03/2015 18:42	12/03/2015 18:42
Service Desk		valid	12/03/2015 19:34	12/03/2015 19:34

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.

An overview of all roles in the system is displayed. To edit a role's settings, click on the role's name. In a fresh new OTRS installation, there are no roles defined by default. To register one, click on the "Add role" button, provide the needed data and submit it (see figure below).

Abbildung 4.9. Adding a new role



Role Management

Actions
Go to overview

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

Add Role

★ Name: IT Supervisor
Validity: valid
Comment: Supervisor of the IT Department

Submit or Cancel

To get an overview of all roles and agents in the system, click on the link Roles <-> Agents on the Admin page. You can also use filters to find a specific element. If you want to change the roles associated with an agent, just click on the agent's name (see figure below). To change the agents associated with a role, click on the role you want to edit (see figure below).

Abbildung 4.10. Change the roles associated with an agent



Manage Role-Agent Relations

Actions
Go to overview

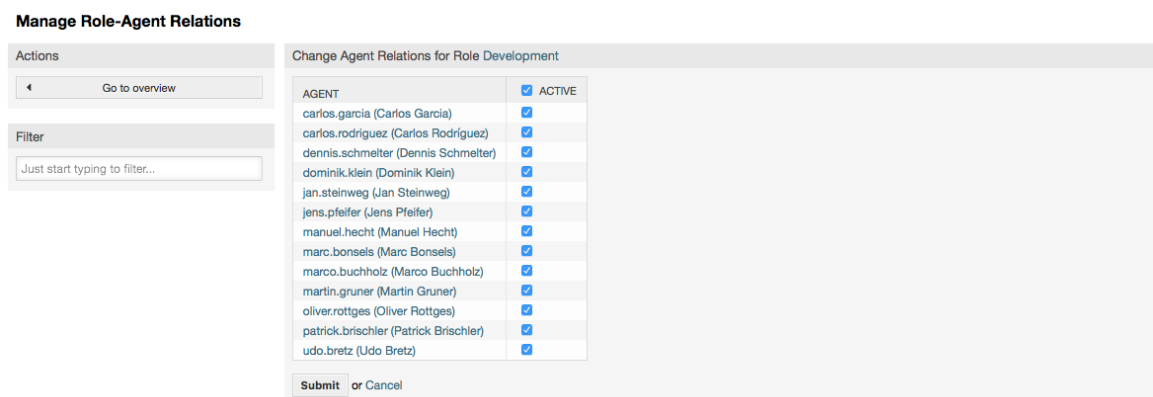
Filter
Just start typing to filter...

Change Role Relations for Agent Jan Steinweg (jan.steinweg)

ROLE	ACTIVE
Development	<input checked="" type="checkbox"/>
IT Supervisor	<input type="checkbox"/>
Service Desk	<input type="checkbox"/>

Submit or Cancel

Abbildung 4.11. Change the agents associated with a specific role



Manage Role-Agent Relations

Actions
Go to overview

Filter
Just start typing to filter...

Change Agent Relations for Role Development

AGENT	ACTIVE
carlos.garcia (Carlos Garcia)	<input checked="" type="checkbox"/>
carlos.rodrguez (Carlos Rodríguez)	<input checked="" type="checkbox"/>
dennis.schmelter (Dennis Schmelter)	<input checked="" type="checkbox"/>
dominik.klein (Dominik Klein)	<input checked="" type="checkbox"/>
jan.steinweg (Jan Steinweg)	<input checked="" type="checkbox"/>
jens.pfeifer (Jens Pfeifer)	<input checked="" type="checkbox"/>
manuel.hecht (Manuel Hecht)	<input checked="" type="checkbox"/>
marc.bonsels (Marc Bonsels)	<input checked="" type="checkbox"/>
marco.buchholz (Marco Buchholz)	<input checked="" type="checkbox"/>
martin.gruner (Martin Gruner)	<input checked="" type="checkbox"/>
oliver.rottgess (Oliver Rottgess)	<input checked="" type="checkbox"/>
patrick.brischler (Patrick Brischler)	<input checked="" type="checkbox"/>
udo.bretz (Udo Bretz)	<input checked="" type="checkbox"/>

Submit or Cancel

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

Abbildung 4.12. Manage roles-groups relations

Manage Role-Group Relations



To define the different access rights for a role, click on the name of a role or a group (see below the Figures 4.13 and 4.14, respectively).

Abbildung 4.13. Change group relations for a role

Manage Role-Group Relations

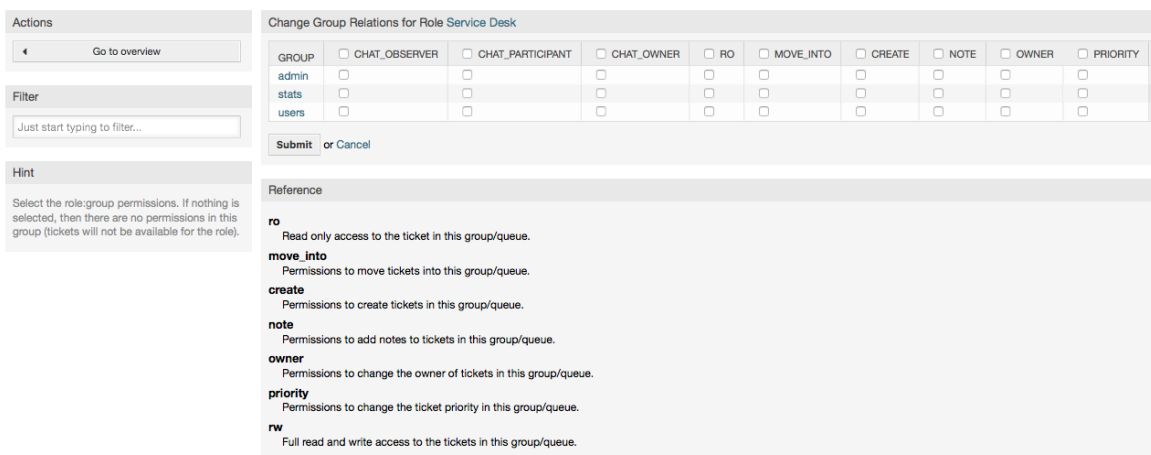
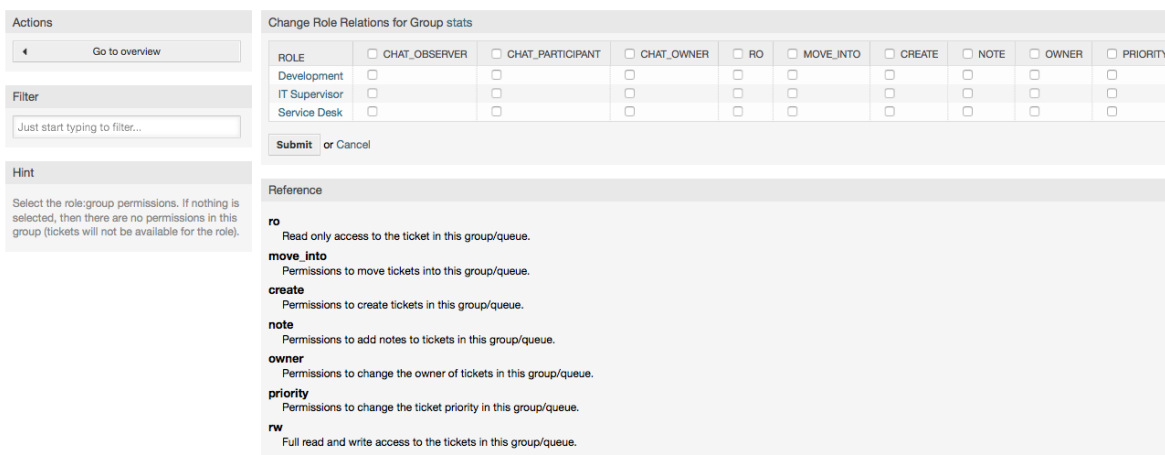


Abbildung 4.14. Change role relations for a group

Manage Role-Group Relations



1.3. Kundenbenutzer und Kundengruppen

1.3.1. Kunden

OTRS supports different types of users. Using the link "Customers" (via the navigation bar, or the Admin page), you can manage the accounts of your customers (see figure below), who can log into the system via the Customers interface (customer.pl). Through this interface, your customers can not only create tickets but also review their past tickets

for new updates. It is important to know that a customer is needed for the ticket history in the system.

Abbildung 4.15. Kundenverwaltung

Customer User Management

Actions

Wildcards like "*" are allowed.

List

USERNAME	NAME	EMAIL	CUSTOMERID	LAST LOGIN	VALIDITY
han.solo	Mr. Han Solo	han.solo@testcustomer.com	SWVII		valid
kylo.ren	Mr. Kylo Ren	kylo.ren@testcustomer.com	SWVII		valid
luke.skywalker	Mr. Luke Skywalker	luke.skywalker@testcustomer.com	SWVII		valid
poe.dameron	Mr. Poe Dameron	poe.dameron@testcustomer.com	SWVII		valid

Hint

Customer user are needed to have a customer history and to login via customer panel.

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.

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

Abbildung 4.16. Adding a customer

Customer User Management

Actions

Hint

Customer user are needed to have a customer history and to login via customer panel.

Add Customer User

Title:

★ **Firstname:**

★ **Lastname:**

★ **Username:**

Password:

★ **Email:**

★ **CustomerID:**

Phone:

Fax:

Mobile:

Street:

Zip:

City:

Country:

Comment:

★ **Valid:**

Interface language:

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

Anmerkung

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

1.3.2. 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

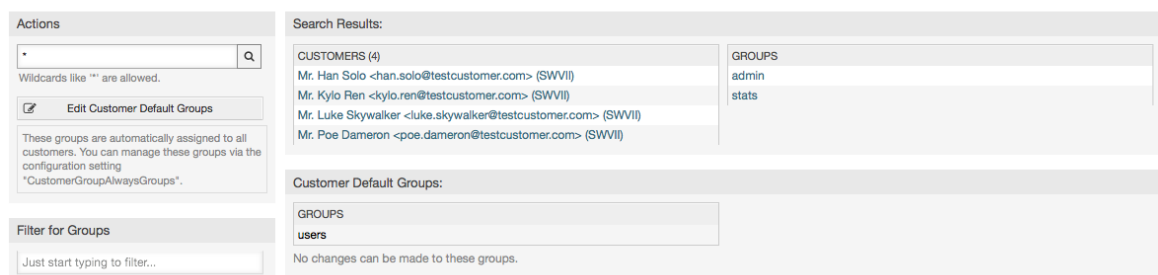
der Gruppenverwaltung an. Fügen Sie dann die Queues hinzu und ordnen Sie diese der neuen Gruppe zu.

The next step is to activate the customer group support. This can be done with the configuration parameter `CustomerGroupSupport`, from the Admin SysConfig option. Using the parameter `CustomerGroupAlwaysGroups`, you can specify the default groups for a newly added customer, so that every new account will be automatically added to these groups.

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

Abbildung 4.17. Customer-Group relations management

Manage Customer-Group Relations



Actions

Wildcards like "*" are allowed.

[Edit Customer Default Groups](#)

These groups are automatically assigned to all customers. You can manage these groups via the configuration setting "CustomerGroupAlwaysGroups".

Filter for Groups

Just start typing to filter...

Search Results:

CUSTOMERS (4)	GROUPS
Mr. Han Solo <han.solo@testcustomer.com> (SWVII)	admin
Mr. Kylo Ren <kylo.ren@testcustomer.com> (SWVII)	stats
Mr. Luke Skywalker <luke.skywalker@testcustomer.com> (SWVII)	
Mr. Poe Dameron <poe.dameron@testcustomer.com> (SWVII)	

Customer Default Groups:

GROUPS

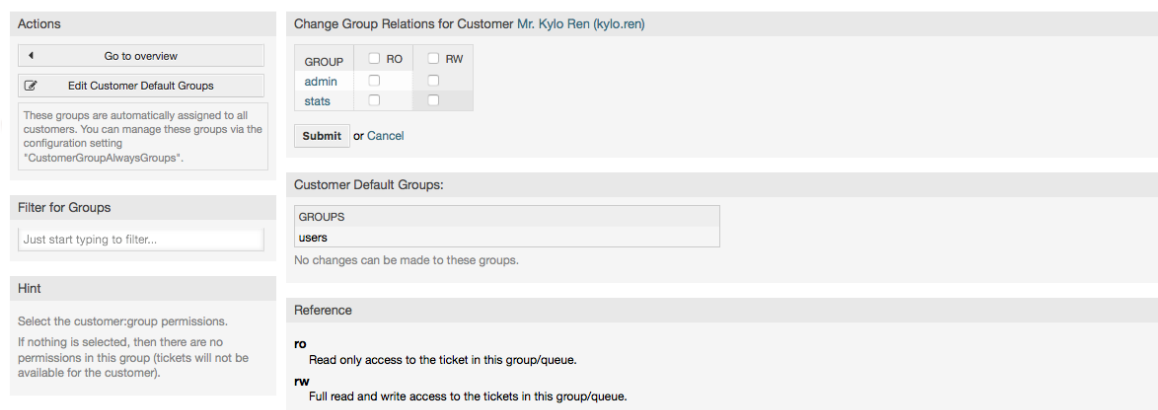
users

No changes can be made to these groups.

To define the different groups a customer should be part of and vice versa, click on the corresponding customer username or group (see below the Figures 4.18 and 4.19, respectively).

Abbildung 4.18. Gruppenzuordnungen eines Kunden bearbeiten

Manage Customer-Group Relations



Actions

[Go to overview](#)

[Edit Customer Default Groups](#)

These groups are automatically assigned to all customers. You can manage these groups via the configuration setting "CustomerGroupAlwaysGroups".

Filter for Groups

Just start typing to filter...

Hint

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

Change Group Relations for Customer Mr. Kylo Ren (kylo.ren)

GROUP	<input type="checkbox"/> RO	<input type="checkbox"/> RW
admin	<input type="checkbox"/>	<input type="checkbox"/>
stats	<input type="checkbox"/>	<input type="checkbox"/>

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

Customer Default Groups:

GROUPS

users

No changes can be made to these groups.

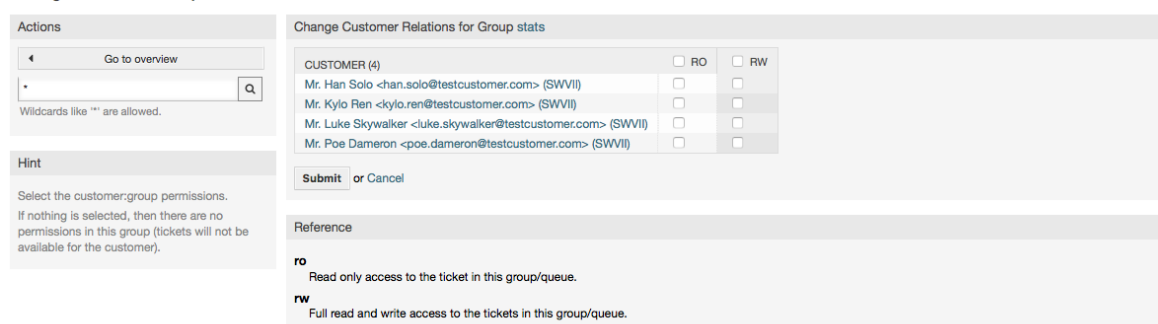
Reference

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

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

Abbildung 4.19. Kundenzuordnungen einer Gruppe bearbeiten

Manage Customer-Group Relations



Actions

[Go to overview](#)

Wildcards like "*" are allowed.

Hint

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

Change Customer Relations for Group stats

CUSTOMER (4)	<input type="checkbox"/> RO	<input type="checkbox"/> RW
Mr. Han Solo <han.solo@testcustomer.com> (SWVII)	<input type="checkbox"/>	<input type="checkbox"/>
Mr. Kylo Ren <kylo.ren@testcustomer.com> (SWVII)	<input type="checkbox"/>	<input type="checkbox"/>
Mr. Luke Skywalker <luke.skywalker@testcustomer.com> (SWVII)	<input type="checkbox"/>	<input type="checkbox"/>
Mr. Poe Dameron <poe.dameron@testcustomer.com> (SWVII)	<input type="checkbox"/>	<input type="checkbox"/>

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

Reference

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

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

1.4. Queues

Clicking on the link "Queues" of the Admin page, you can manage the queues of your system (see figure below). In a new OTRS installation there are 4 default queues: Raw, Junk, Misc and Postmaster. All incoming messages will be stored in the "Raw" queue if no filter rules are defined. The "Junk" queue can be used to store spam messages.

Abbildung 4.20. Queue-Verwaltung

Manage Queues

NAME	GROUP	COMMENT	VALIDITY	CHANGED	CREATED
Junk	users	All junk tickets.	valid	11/19/2015 13:25	11/19/2015 13:25
Misc	users	All misc tickets.	valid	11/19/2015 13:25	11/19/2015 13:25
Postmaster	users	Postmaster queue.	valid	11/19/2015 13:25	11/19/2015 13:25
Raw	users	All default incoming ti...	valid	11/19/2015 13:25	11/19/2015 13:25

Here you can add queues (see figure below) and modify them. You can specify the group that should use the queue. You can also set the queue as a sub-queue of an existing queue.

Abbildung 4.21. Adding a new queue

Manage Queues

Add Queue

★ Name:

Sub-queue of:

★ Group:

Unlock timeout minutes:

0 = no unlock - 24 hours = 1440 minutes - Only business hours are counted.
 If an agent locks a ticket and does not close it before the unlock timeout has passed, the ticket will unlock and will become available for other agents.

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

0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.
 If there is not added a customer contact, either email-external or phone, to a new ticket before the time defined here expires, the ticket is escalated.

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

0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.
 If there is an article added, such as a follow-up via email or the customer portal, the escalation update time is reset. If there is no customer contact, either email-external or phone, added to a ticket before the time defined here expires, the ticket is escalated.

Escalation - solution time (minutes): (Notify by)

0 = no escalation - 24 hours = 1440 minutes - Only business hours are counted.
 If the ticket is not set to closed before the time defined here expires, the ticket is escalated.

★ Follow up Option:

Specifies if follow up in closed tickets would re-open the ticket, be related or lead to a new ticket

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

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

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.

1.5. Salutations, Signatures, Attachments and Templates

1.5.1. Anreden

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

Abbildung 4.22. Verwaltung der Anreden

Salutation Management

Actions		List				
<input type="button" value="Add salutation"/>		NAME	COMMENT	VALIDITY	CHANGED	CREATED
		system standard salutation (en)	Standard Salutation.	valid	11/19/2015 13:25	11/19/2015 13:25

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

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

Abbildung 4.23. Adding a new salutation

Salutation Management

Actions: [Go to overview](#)

Add Salutation

★ Name:

Salutation:

B I U S [Rich Text Editor Icons]

Format - Font - Size - [Color Picker] [Background Color Picker] [Link] [Source] [Ω] [↻]

Hello <OTRS_CUSTOMER_DATA_UserFirstname>,
 We are working on your ticket <OTRS_TICKET_TicketNumber>
 ...
 ...
 ...

★ Validity:

Comment:

or

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

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

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.

1.5.2. Signaturen

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

Abbildung 4.24. Signatures management

Signature Management

Actions:

NAME	COMMENT	VALIDITY	CHANGED	CREATED
system standard signature (en)	Standard Signature.	valid	11/19/2015 13:25	11/19/2015 13:25

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

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

Abbildung 4.25. Adding a new signature

Signature Management

Actions

Go to overview

Add Signature

★ Name: Signature example

Signature:

B I U S [List Bullets] [List Numbers] [List Symbols] [List Links] [List Images] [List Videos] [List Audio] [List Code] [List Source] [List Omega] [List Print] [List Refresh]

Format - Font - Size - A - [Image] - I_x [Image] Source Ω [Image] [Image]

It is our pleasure to serve you.

Receive kind greetings,
 <OTRS_RESPONSIBLE_UserFirstname> <OTRS_RESPONSIBLE_UserLastname>

★ Validity: valid

Comment:

Submit or Cancel

Hint

Auch in Anreden und 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. Wenn Sie zum Beispiel die Variable `<OTRS_LAST_NAME>` in einer Signatur hinzufügen, wird der Name des Mitarbeiters angezeigt, der das Ticket beantwortet hat.

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.

1.5.3. Anlagen

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

Through the "Attachment" link of the Admin page, you can load the attachments into the database of the system (see figure below).

Abbildung 4.26. Attachments management

Attachment Management

Actions

Add attachment

List

NAME	FILENAME	COMMENT	VALIDITY	CHANGED	CREATED	DELETE
Sample One	mar.jpg		valid	12/03/2015 18:59	12/03/2015 18:59	
Sample Two	download.jpg		valid	12/03/2015 18:59	12/03/2015 18:59	

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

Abbildung 4.27. Adding a new attachment

Attachment Management

Actions
 ◀ Go to overview

Add Attachment

★ Name:

★ Attachment: No file selected.

★ Validity:

Comment:

or

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

Abbildung 4.28. Anhänge zu Vorlagen zuordnen.

Manage Templates <-> Attachments Relations

Filter for Templates

Filter for Attachments

Overview

TEMPLATES	ATTACHMENTS
Answer - A new answer	Sample One (mar.jpg)
Answer - empty answer	Sample Two (download.jpg)
Answer - test answer	
Create - Create one	
Forward - Forward one	
Note - Note one	

To associate different attachments with a specific template and vice versa, click on the corresponding template name or attachment (see below the Figures 4.29 and 4.30, respectively).

Abbildung 4.29. Anlagenzuordnungen einer Vorlage verändern

Manage Templates <-> Attachments Relations

Actions
 ◀ Go to overview

Filter

Change Attachment Relations for Template Answer - empty answer

ATTACHMENT	<input type="checkbox"/> ACTIVE
Sample One (mar.jpg)	<input type="checkbox"/>
Sample Two (download.jpg)	<input type="checkbox"/>

or

Abbildung 4.30. Vorlagen-Zuordnungen für einen Anhang verändern

Manage Templates <-> Attachments Relations

Actions
 ◀ Go to overview

Filter

Change Template Relations for Attachment Sample Two

TEMPLATE	<input type="checkbox"/> ACTIVE
Answer - A new answer	<input type="checkbox"/>
Answer - empty answer	<input type="checkbox"/>
Answer - test answer	<input type="checkbox"/>
Create - Create one	<input type="checkbox"/>
Forward - Forward one	<input type="checkbox"/>
Note - Note one	<input type="checkbox"/>

or

1.5.4. Vorlagen

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

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

- Antwort: Als Ticketantwort oder Rückantwort verwendet.
- Erstellen: In neuen Telefon- oder E-Mailtickets verwendet.
- Weiterleiten: Verwendet um einen Artikel an jemand Anderen weiterzuleiten.
- Telefonanruf: In Telefonanruf eingehend und ausgehend verwendet.

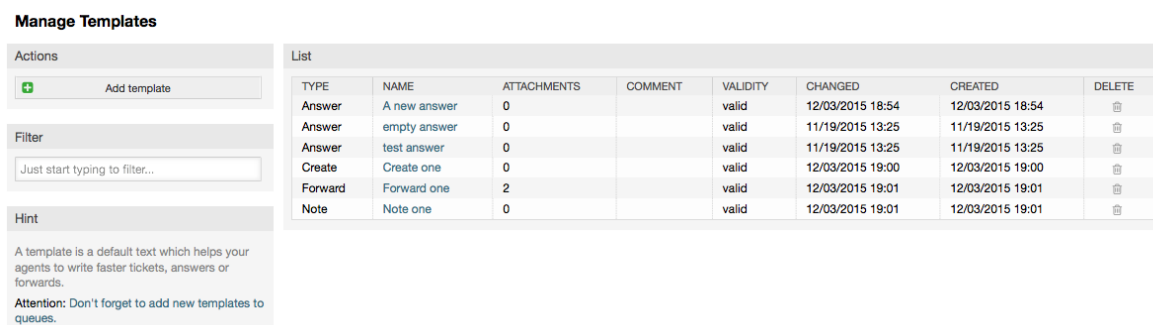
Answer templates can be accessed in two ways, from the ticket zoom screen in the article menu, or on a quicker way: from any ticket overview large screen such as Status View or Ticket View. For a fresh OTRS installation, the "empty answer" template (Answer) is set as the default for every queue.

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

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

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

Abbildung 4.31. Template management



Manage Templates

Actions

Filter

Hint

A template is a default text which helps your agents to write faster tickets, answers or forwards.
Attention: Don't forget to add new templates to queues.

TYPE	NAME	ATTACHMENTS	COMMENT	VALIDITY	CHANGED	CREATED	DELETE
Answer	A new answer	0		valid	12/03/2015 18:54	12/03/2015 18:54	
Answer	empty answer	0		valid	11/19/2015 13:25	11/19/2015 13:25	
Answer	test answer	0		valid	11/19/2015 13:25	11/19/2015 13:25	
Create	Create one	0		valid	12/03/2015 19:00	12/03/2015 19:00	
Forward	Forward one	2		valid	12/03/2015 19:01	12/03/2015 19:01	
Note	Note one	0		valid	12/03/2015 19:01	12/03/2015 19:01	

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

Abbildung 4.32. Adding a template

Manage Templates

Actions

Hint
 A template is a default text which helps your agents to write faster tickets, answers or forwards.
Attention: Don't forget to add new templates to queues.

Add Template

★ Type:

★ Name:

Template:

B I U S [List Icons] [Align Icons] [Text Color] [Background Color] [Source] [Ω] [↶] [↷] [Q]

Format - Font - Size - [A] [A] [I_x] [Source] [Ω] [↶] [↷]

The text for this answer . . .

Attachments:

★ Validity:

Comment:

or

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

Abbildung 4.33. Template-Queue relations management

Manage Template-Queue Relations

Filter for Templates

Filter for Queues

Overview

TEMPLATES	QUEUES
Answer - A new answer	Junk
Answer - empty answer	Misc
Answer - test answer	Postmaster
Create - Create one	Raw
Forward - Forward one	
Note - Note one	

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

Abbildung 4.34. Ändern der Queue Zuordnung für eine Vorlage

Manage Template-Queue Relations

Actions

Filter

Change Queue Relations for Template Answer - empty answer

QUEUE	<input type="checkbox"/> ACTIVE
Junk	<input checked="" type="checkbox"/>
Misc	<input checked="" type="checkbox"/>
Postmaster	<input checked="" type="checkbox"/>
Raw	<input checked="" type="checkbox"/>

or

Abbildung 4.35. Antwort-Zuordnungen einer Queue ändern

Manage Template-Queue Relations

Actions

Filter

Change Template Relations for Queue Junk

TEMPLATE	<input type="checkbox"/> ACTIVE
Answer - A new answer	<input type="checkbox"/>
Answer - empty answer	<input checked="" type="checkbox"/>
Answer - test answer	<input type="checkbox"/>
Create - Create one	<input type="checkbox"/>
Forward - Forward one	<input type="checkbox"/>
Note - Note one	<input type="checkbox"/>

or

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

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

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

Die Weiterleiten Vorlagen sind ähnlich zu den Antwort Vorlagen, enthalten aber keine Anrede.

1.6. Automatische Antworten

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

Abbildung 4.36. Auto response management

Auto Response Management

NAME	TYPE	COMMENT	VALIDITY	CHANGED	CREATED
default follow-up (after a ticket follow-up has been added)	auto follow up		valid	11/19/2015 13:25	11/19/2015 13:25
default reject (after follow-up and rejected of a closed ticket)	auto reject		valid	11/19/2015 13:25	11/19/2015 13:25
default reject/new ticket created (after closed follow-up with new ticket creation)	auto reply/new ticket		valid	11/19/2015 13:25	11/19/2015 13:25
default reply (after new ticket has been created)	auto reply		valid	11/19/2015 13:25	11/19/2015 13:25

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

Abbildung 4.37. Adding an auto response

Auto Response Management

Actions: [Go to overview](#)

★ Name:

★ Subject:

Response:

Format | Font | Size |

Hello <OTRS_CUSTOMER_REALNAME>

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

You will have news in the next 24 hours.

Thank you!

Your OTRS Team

★ Type:

★ Auto response from:

★ Validity:

Comment:

or

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.

For every automatic answer, you can specify the event that should trigger it. The system events that are available after a default installation are described in the Table 4-4.

Tabelle 4.4. Events for auto responses

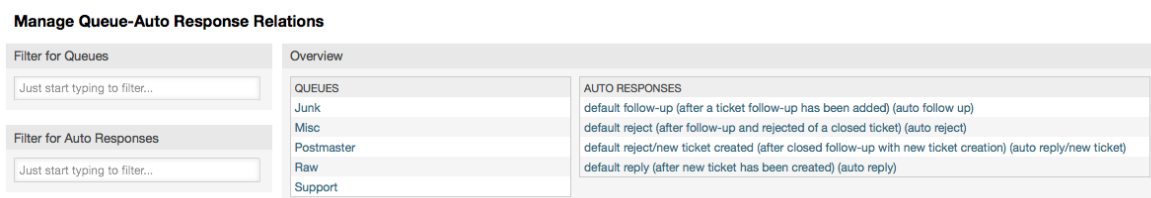
Name	Beschreibung
auto reply	Dieses Ereignis tritt ein, wenn ein neues Ticket in einer Queue angelegt wird.
auto reply/new ticket	Dieses Ereignis tritt ein, wenn ein bereits geschlossenes Ticket, z. B. durch die Antwort eines Kunden, mit einer neuen Ticketnummer erneut geöffnet wird.
auto follow up	Dieses 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

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

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

Abbildung 4.38. Queue <-> auto response relations management



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

Abbildung 4.39. Change auto response relations for a queue

Manage Queue-Auto Response Relations

Filter for Queues

Overview

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

Filter for Auto Responses

1.7. System Email Addresses

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

Abbildung 4.40. System email addresses management

System Email Addresses Management

Actions

[Add system address](#)

Hint

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

List

EMAIL ADDRESS	DISPLAY NAME	QUEUE	VALIDITY	CHANGED	CREATED
otrs@localhost	OTRS System	Postmaster	valid	11/19/2015 13:25	11/19/2015 13:25
postmaster@mycompany.com	Postmaster	Junk	valid	01/03/2016 19:16	01/03/2016 19:16
support@mycompany.com	Support Team	Junk	valid	01/03/2016 19:15	01/03/2016 19:15

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

Abbildung 4.41. Adding a system email address

System Email Addresses Management

Actions

[Go to overview](#)

Hint

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

Add System Email Address

★ Email address:

★ Display name:

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

★ Queue:

★ Validity:

Comment:

or

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.

1.8. Ticket-Benachrichtigungen

OTRS allows ticket notifications to be sent to agents and customers, based on the occurrence of certain events. Agents can customize their ticket notification settings via the preferences link.

Through the "Ticket Notifications" link on the Admin page, you can manage the ticket notifications of your system (see figure below). OTRS comes with a set of predefined notifications that cover a wide range of use cases.

Abbildung 4.42. Ticket notification management

Ticket Notification Management

Actions

Configuration Import

Here you can upload a configuration file to import Ticket Notifications to your system. The file needs to be in .yaml format as exported by the Ticket Notification module.

No file selected.

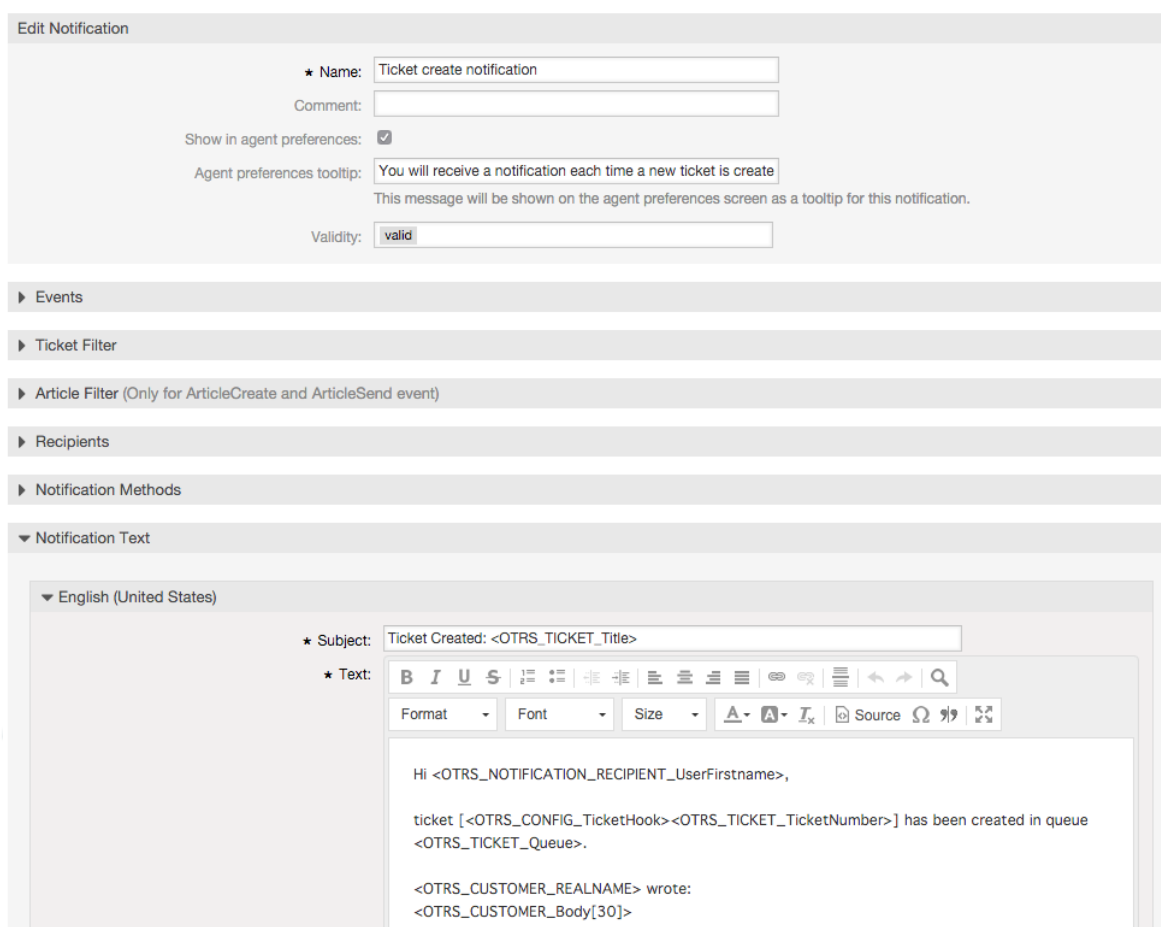
Overwrite existing notifications?

List

NAME	COMMENT	VALIDITY	CHANGED	CREATED	EXPORT	COPY	DELETE
Ticket create notification		valid	11/19/2015 13:25	11/19/2015 13:25			
Ticket escalation notification		valid	11/19/2015 13:25	11/19/2015 13:25			
Ticket escalation warning notification		valid	11/19/2015 13:25	11/19/2015 13:25			
Ticket follow-up notification (locked)		valid	11/19/2015 13:25	11/19/2015 13:25			
Ticket follow-up notification (unlocked)		valid	11/19/2015 13:25	11/19/2015 13:25			
Ticket lock timeout notification		valid	11/19/2015 13:25	11/19/2015 13:25			
Ticket new note notification		valid	11/19/2015 13:25	11/19/2015 13:25			
Ticket owner update notification		valid	11/19/2015 13:25	11/19/2015 13:25			
Ticket pending reminder notification (locked)		valid	11/19/2015 13:25	11/19/2015 13:25			
Ticket pending reminder notification (unlocked)		valid	11/19/2015 13:25	11/19/2015 13:25			
Ticket queue update notification		valid	11/19/2015 13:25	11/19/2015 13:25			
Ticket responsible update notification		valid	11/19/2015 13:25	11/19/2015 13:25			
Ticket service update notification		valid	11/19/2015 13:25	11/19/2015 13:25			

You can customize many aspects of the notifications. Click on the notification you want to change, and its content will be loaded for editing (see figure below).

Abbildung 4.43. Customizing a notification



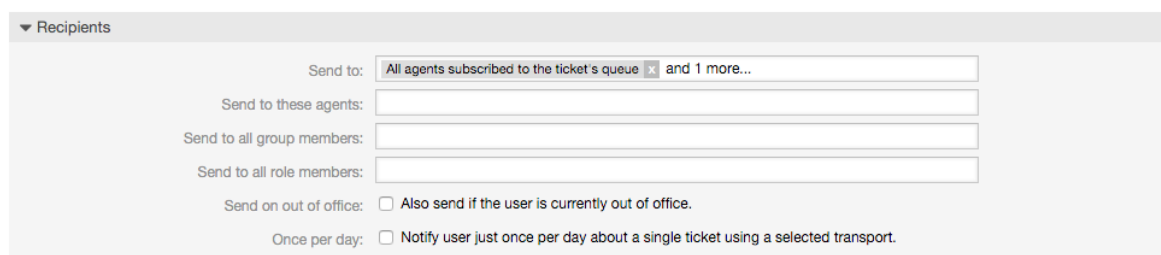
You can edit the basic data of this notification such as name and comment, and control if the agents may choose to receive this notification (per transport method). For every language, a subject and body can be added/edited to configure what will actually be sent as the notification content.

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

You can choose which events should trigger this notification, and limit it to tickets which match certain criteria (ticket and/or article filter). This makes it possible to create different notifications for different queues, priorities or other criteria that might be relevant for your system.

The recipients of the notification can be configured according to different criteria (groups, roles, individual agents etc.). All configured recipients will receive the notification.

Abbildung 4.44. Customizing a notification's recipients



Additionally, you can specify if the notification should be sent to agents who are out of office, and limit the sending to once per day and ticket (e. g. pending reminder notification).

Notifications can be sent with different notification methods. The "Email" notification method is available in OTRS Free, with **OTRS Business Solution™** you also get the possibility to store and view the notifications in the database (so that no email client is needed to use OTRS) as well as to send them via SMS (e. g. for very important notifications).

Abbildung 4.45. Customizing notification methods

▼ Notification Methods

These are the possible methods that can be used to send this notification to each of the recipients. Please select at least one method below.

Email

Enable this notification method:

Active by default in agent preferences:
This is the default value for assigned recipient agents who didn't make a choice for this notification in their preferences yet. If the box is enabled, the notification will be sent to such agents.

Additional recipient email addresses:

Notification article type:
An article will be created if the notification is sent to the customer or an additional email address.

Email template:
Use this template to generate the complete email (only for HTML emails).

Enable email security:
PGP and SMIME not enabled.

Email security level:

If signing key/certificate is missing:

If encryption key/certificate is missing:

Web View

Enable this notification method:

Active by default in agent preferences:
This is the default value for assigned recipient agents who didn't make a choice for this notification in their preferences yet. If the box is enabled, the notification will be sent to such agents.

SMS (Short Message Service)

Enable this notification method:

Active by default in agent preferences:
This is the default value for assigned recipient agents who didn't make a choice for this notification in their preferences yet. If the box is enabled, the notification will be sent to such agents.

Recipient SMS numbers:

1.9. S/MIME

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

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

Abbildung 4.46. S/MIME management

S/MIME Management:

Actions

+ Add certificate

+ Add private key

Filter for certificates

Just start typing to filter...

Hint

To show certificate details click on a certificate icon.
To manage private certificate relations click on a private key icon.

Results

	TYPE	SUBJECT	HASH	FINGERPRINT	CREATE	EXPIRES	DELETE
No data found.							

1.10. PGP

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

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

Abbildung 4.47. PGP management

PGP Management

Actions

Search:

+ Add PGP key

Hint

In this way you can directly edit the keyring configured in SysConfig.
Description: Introduction to PGP

Result

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

1.11. Status

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

Abbildung 4.48. State management

State Management

Actions

+ Add state

Hint

Attention: Please also update the states in SysConfig where needed.
See also: <http://otrs.github.io/doc>

List

NAME	TYPE	COMMENT	VALIDITY	CHANGED	CREATED
closed successful	closed	Ticket is closed ...	valid	11/19/2015 13:25	11/19/2015 13:25
closed unsuccessful	closed	Ticket is closed ...	valid	11/19/2015 13:25	11/19/2015 13:25
merged	merged	State for merged ...	valid	11/19/2015 13:25	11/19/2015 13:25
new	new	New ticket create...	valid	11/19/2015 13:25	11/19/2015 13:25
open	open	Open tickets.	valid	11/19/2015 13:25	11/19/2015 13:25
pending auto close+	pending auto	Ticket is pending...	valid	11/19/2015 13:25	11/19/2015 13:25
pending auto close-	pending auto	Ticket is pending...	valid	11/19/2015 13:25	11/19/2015 13:25
pending reminder	pending reminder	Ticket is pending...	valid	11/19/2015 13:25	11/19/2015 13:25
removed	removed	Customer removed ...	valid	11/19/2015 13:25	11/19/2015 13:25

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

1.12. Die SysConfig (Verwaltung der Systemkonfiguration)

In der SysConfig werden die meisten Konfigurationseinstellungen von OTRS vorgenommen.

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

Abbildung 4.49. Das grafische Konfigurationsfrontend von OTRS (SysConfig)

SysConfig

Actions

Navigate by searching in 1574 settings

Navigate by selecting config groups

Export settings

Import settings

Result

SUBGROUP	ELEMENTS	GROUP
Core	33	Framework
Core::Cache	4	Framework
Core::CustomerCompany	1	Framework
Core::CustomerUser	2	Framework
Core::Fetchmail	1	Framework
Core::LinkObject	4	Framework
Core::Log	7	Framework
Core::MIME-Viewer	4	Framework
Core::MirrorDB	8	Framework
Core::OTRSBusiness	1	Framework
Core::PDF	11	Framework
Core::Package	8	Framework
Core::PerformanceLog	3	Framework
Core::ReferenceData	1	Framework
Core::SOAP	3	Framework
Core::Sendmail	10	Framework
Core::Session	18	Framework
Core::SpellChecker	4	Framework
Core::Stats	3	Framework
Core::Time	10	Framework
Core::Time::Calendar1	6	Framework
Core::Time::Calendar2	6	Framework
Core::Time::Calendar3	6	Framework
Core::Time::Calendar4	6	Framework

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

1.13. Using Mail Accounts

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.

Abbildung 4.50. Verwaltung von Mail-Konten

Mail Account Management

Actions

+ Add mail account

Hint

All incoming emails with one account will be dispatched in the selected queue!

If your account is trusted, the already existing X-OTRS header at arrival time (for priority, ...) will be used! PostMaster filter will be used anyway.

List

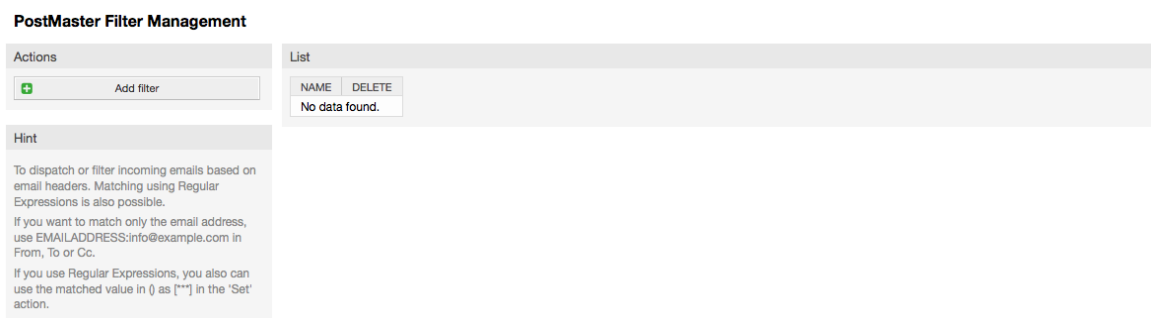
HOST/USERNAME	TYPE	COMMENT	VALIDITY	CHANGED	CREATED	DELETE	RUN NOW!
No data found.							

Nähere Informationen erhalten Sie im Kapitel Postmaster Mailaccounts.

1.14. Filtering Incoming Email Messages

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

Abbildung 4.51. PostMaster filter management



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

Anmerkung

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

Tabelle 4.5. 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	Sets the pending time of a ticket (you also should sent a pending state via X-OTRS-State). You can specify absolute dates like "2010-11-20 00:00:00" or relative dates, based on the arrival time of the email. Use the form "+ \$Number \$Unit", where \$Unit can

Name	Mögliche Werte	Beschreibung
		be 's' (seconds), 'm' (minutes), 'h' (hours) or 'd' (days). Only one unit can be specified. Examples of valid settings: "+50s" (pending in 50 seconds), "+30m" (30 minutes), "+12d" (12 days). Note that settings like "+1d 12h" are not possible. You can specify "+36h" instead.
X-OTRS-Type:	default (systemabhängig)	Setzt den Ticket-Typ (wenn Ticket::Type aktiviert ist).
X-OTRS-Service:	(systemabhängig)	Setzt den Service eines Tickets (wenn Ticket::Service aktiviert ist). Wenn Sie einen Sub-Service erstellen möchten, müssen Sie einen vorhandenen Service als "Parent::Sub" wählen.
X-OTRS-SLA:	(systemabhängig)	Setzt das Service-Level-Abkommen eines Tickets (wenn Ticket::Service aktiviert ist).
X-OTRS-CustomerUser:	CustomerUser	Legt den Kunden-Benutzer fest, dem das Ticket zugeordnet werden soll.
X-OTRS-CustomerNo:	CustomerNo	Legt die Kunden-ID fest, die dem Ticket zugeordnet werden soll.
X-OTRS-SenderType:	agent, system, customer	Legt die Art des Ticket-Absenders fest.
X-OTRS-ArticleType:	email-external, email-internal, email-notification-ext, email-notification-int, phone, fax, sms, webrequest, note-internal, note-external, note-report	Legt den Typ des Artikles für das eingehende Ticket fest.
X-OTRS-DynamicField-<DynamicFieldName>:	Hängt von der Konfiguration der dynamischen Felder ab (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 Mail-schleifen 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).

Filter rules are evaluated in alphabetical order, and are all executed except if the "Stop after match" setting has been set to "Yes" in one of the rules (in this case evaluation of the remaining filters is canceled).

Abbildung 4.52. Add a PostMaster filter

PostMaster Filter Management

Actions

Go to overview

Hint

To dispatch or filter incoming emails based on email headers. Matching using Regular Expressions is also possible.

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

If you use Regular Expressions, you also can use the matched value in {} as ["{}"] in the 'Set' action.

Add PostMaster Filter

Name:

Stop after match:

Filter Condition (AND Condition)

Check email header: Negate: Look for value:

Check email header: Negate: Look for value:

Check email header: Negate: Look for value:

Check email header: Negate: Look for value:

Beispiel 4.1. Aussortierung von Spammails in eine bestimmte Queue

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

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

1.15. Executing Automated Jobs with the GenericAgent

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

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

Abbildung 4.53. Jobliste des GenericAgent

Generic Agent

Actions

Add job

List

NAME	LAST RUN	VALIDITY	DELETE	RUN NOW!
Job One		valid	Delete	Run this task
Job Three		valid	Delete	Run this task
Job Two		valid	Delete	Run this task

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

Abbildung 4.54. Anlegen eines GenericAgent-Jobs

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

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

Warnung

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

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

1.16. Administrative Messages

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

Abbildung 4.55. Admin notification screen

Admin Notification

Hint
With this module, administrators can send messages to agents, group or role members.

Create Administrative Message

★ From: admin@mycompany.com

Send message to users: marc.bonsels x martin.gruner x and 1 more...

Send message to group members:

Group members need to have permission: ro rw

Send message to role members: Service Desk x

Also send to customers in groups:

★ Subject: A complete subject for notification

★ Body: **B I U S** [Rich Text Editor]
Text example . . .

Send

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.

1.17. Sitzungsverwaltung

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

Abbildung 4.56. Sitzungsverwaltung

Session Management

Actions	List
All sessions 2	SESSION
Agent sessions 1	3sydOnqpwHQLLxa4083rVcTrmBC1wPF8
Customer sessions 1	jzBgJlquUNb6950CaGHAN0tsME7eA5Fn
Unique agents 1	
Unique customers 1	
Kill all sessions	

SESSION	TYPE	USER	KILL
3sydOnqpwHQLLxa4083rVcTrmBC1wPF8	Agent	Carlos Garcia	Kill this session
jzBgJlquUNb6950CaGHAN0tsME7eA5Fn	Customer	Han Solo	Kill this session

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

Abbildung 4.57. Sitzungsdetails

Session Management

Actions

← Go to overview

⏻ Kill this session

Detail View for SessionID : 3sydOnqpwHQLUxa4083rVcTrmBC1wP18 - Carlos Garcia

KEY	VALUE
AdminDynamicFieldsOverviewPageShown	25
ChangeTime	2015-11-25 13:25:36
CreateTime	2015-11-25 13:25:36
ExternalChannels	[]
NotificationTransport	{"Notification-3-Email";0,"Notification-1-Email";0,"Notification-4-Email";0,"Notification-8-Email";0,"Notification-13-Email";0,"Notification-2-Email";0}
OutOfOfficeEndDay	12
OutOfOfficeEndMonth	12
OutOfOfficeEndYear	2015
OutOfOfficeStartDay	11
OutOfOfficeStartMonth	12
OutOfOfficeStartYear	2015
SessionID	3sydOnqpwHQLUxa4083rVcTrmBC1wP18
UserChallengeToken	4K1P2yUGJdCMXetMeSFIZom4zCajRRfh
UserEmail	carlos.garcia@mycompany.com
UserFirstname	Carlos
UserFullname	Carlos Garcia
UserID	2
UsersGroupRo[admin]	Yes
UsersGroupRo[stats]	Yes
UsersGroupRo[users]	Yes
UsersGroup[admin]	Yes
UsersGroup[stats]	Yes

1.18. Systemwartung

System Maintenance give the option to schedule one or more maintenance periods for the system. During this period no agents or customers can login into the system (except for Agents in the "admin" group). Current logged users and customers receive a notification about the maintenance (before and during the maintenance period). Administrators have the option to kill the sessions for logged agents and customers, all this in preparation to be able to make changes in the system (e.g. a system update) in a "safe" environment.

Abbildung 4.58. The system maintenance overview screen with some scheduled periods

System Maintenance Management

Actions

+ Schedule New System Maintenance

List

START DATE	STOP DATE	COMMENT	VALIDITY	DELETE
2015-12-03 19:57:00	2015-12-03 22:57:00	A comment about this maintenance period	valid	🗑️

Hint

Schedule a system maintenance period for announcing the Agents and Customers the system is down for a time period.

Some time before this system maintenance starts the users will receive a notification on each screen announcing about this fact.

The Start Date and the Stop Date are required fields, and the only rule for this combination is that Start Date can not be a date after the Stop Date.

Abbildung 4.59. The system maintenance edit screen

Edit System Maintenance

Actions
Go to overview

▼ Edit System Maintenance information

Start date: 12 / 03 / 2015 19 : 57

Stop date: 12 / 03 / 2015 22 : 57

★ Comment: A comment about this maintenance period

Login message: We apologize for the inconveniences, the system will be ready soon.

Show login message:

Notify message: A notification message for customer.

★ Validity: valid

Save or Cancel

▼ Manage Sessions

All Sessions 2
Unique agents 1
Unique customers 1

Agent Sessions

SESSION	TYPE	USER	KILL
3sydOnqpwHQLUxa4083rVcTfmBC1wPF8	Agent	Carlos Garcia	Kill this session

Customer Sessions

SESSION	TYPE	USER	KILL
jzBgJlquUNb6950CaGHAN0tsME7eA5Fn	Customer	Han Solo	Kill this session

Kill all Sessions, except for your own

Nachdem eine neue Wartungszeit angelegt wurde, wird eine Übersicht mit Details über aktive Sessions angezeigt, in der Administratoren einzelne oder alle Sessions (außer der aktuellen) beenden können wenn das benötigt ist.

1.19. System Log

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

Abbildung 4.60. System Log

System Log

Hint
Here you will find log information about your system.
 Hide this message

Recent Log Entries

TIME	PRIORITY	FACILITY	MESSAGE
Tue Dec 2 02:00:12 2014	notice	OTRS-otrs.GenericAgent.pl-2664	Use module (Kernel::System::GenericAgent::TriggerAdvancedEscalationStartEvents) for Ticket (201411122664000012/3).
Tue Dec 2 02:00:11 2014	notice	OTRS-otrs.GenericAgent.pl-2664	Added scheduler job 'EscalationHistory' by escalation event 'EscalationBreachd_2' for ticket '2'!
Tue Dec 2 02:00:11 2014	notice	OTRS-CGI-3051	CustomerUser: 'bruce.banner' changed password successfully!
Tue Dec 2 02:00:11 2014	notice	OTRS-otrs.GenericAgent.pl-1092	Use module (Kernel::System::GenericAgent::TriggerEscalationStartEvents) for Ticket (109200664/990).

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

Anmerkung

System logs are available via the web interface only on Linux / Unix systems.

1.20. SQL Queries via the SQL Box

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

Abbildung 4.61. SQL Box

SQL Box

Hint

Here you can enter SQL to send it directly to the application database. It is not possible to change the content of the tables, only select queries are allowed.

Options

★ SQL:

Limit:

Result format:

1.21. Paketverwaltung

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

Abbildung 4.62. Paketverwaltung

Package Manager

Actions

No file selected.

Online Repository

NAME	VERSION	VENDOR	DESCRIPTION	ACTION
FAQ	5.0.2	OTRS AG	The FAQ/knowledge base.	Install
Fred	3.2.7	OTRS AG	A tool to support the developer by his development.	Install
OTRSCloneDB	5.0.1	OTRS AG	The OTRS CloneDB package.	Install
OTRSCodePolicy	1.0.8	OTRS AG	OTRS code quality checks.	Install
OTRSMasterSlave	5.0.1	OTRS AG	Includes "Ticket Master/Slave" feature.	Install
Survey	5.0.1	OTRS AG	A customer survey tool.	Install
SystemMonitoring	5.0.1	OTRS AG	Basic mail interface to System Monitoring Suites. Al...	Install
TimeAccounting	5.0.1	OTRS AG	A Time Registration Module.	Install

Local Repository

NAME	VERSION	VENDOR	DESCRIPTION	STATUS	ACTION
No data found.					

Features for **OTRS Business Solution™** customers only → sales@otrs.com

With **OTRS Business Solution™**, you can benefit from the following optional features. Please make contact with sales@otrs.com if you need more information.

NAME	DESCRIPTION
Advanced Editor	Makes it possible to use response templates with less resources
Custom Contact Fields	Makes it possible to store customer contact data directly in the ticket
Customer Interface Link Object	Shows linked tickets and FAQ article in the OTRS Customer Portal.
Ticket Workflow ITSM	Define Ticket Workflows especially for working processes in your IT Service Management
Service Categories	Assignment of ticket times to ticket services

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

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

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

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

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

1.22. 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).

Abbildung 4.63. Die grafische Konfigurationsoberfläche für Webservices

GenericInterface Web Service Management - Overview

You are here: > Web Services

Actions

+ Add web service

Web Service List

NAME	DESCRIPTION	REMOTE SYSTEM	PROVIDER TRANSPORT	REQUESTER TRANSPORT	VALIDITY
Webservice one	-	-	-	-	valid
Webservice two	-	-	-	-	valid

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

1.23. Dynamische Felder

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

Abbildung 4.64. The dynamic fields overview screen with some dynamic fields

Dynamic Fields Management - Overview

Actions

Ticket

Add new field for object: Ticket

Article

Add new field for object: Article

Hint

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

Dynamic Fields List

NAME	LABEL	ORDER	TYPE	OBJECT	VALIDITY	DELETE
ProcessManagementProcessID	Process	1	ProcessID	Ticket	valid	
ProcessManagementActivityID	Activity	2	ActivityID	Ticket	valid	

1-2 of 2

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.

2. Systemkonfiguration

2.1. 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.

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`.

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

2.2. Configuring the System Through the Web Interface

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).

Abbildung 4.65. Die grafische System-Konfigurationsoberfläche

SysConfig

Actions

Search:

Navigate by searching in 1574 settings

Navigate by selecting config groups

[Export settings](#)

[Import settings](#)

Result

SUBGROUP	ELEMENTS	GROUP
Core	33	Framework
Core::Cache	4	Framework
Core::CustomerCompany	1	Framework
Core::CustomerUser	2	Framework
Core::Fetchmail	1	Framework
Core::LinkObject	4	Framework
Core::Log	7	Framework
Core::MIME-Viewer	4	Framework
Core::MirrorDB	8	Framework
Core::OTRSBusiness	1	Framework
Core::PDF	11	Framework
Core::Package	8	Framework
Core::PerformanceLog	3	Framework
Core::ReferenceData	1	Framework
Core::SOAP	3	Framework
Core::Sendmail	10	Framework
Core::Session	18	Framework
Core::SpellChecker	4	Framework
Core::Stats	3	Framework
Core::Time	10	Framework
Core::Time::Calendar1	6	Framework
Core::Time::Calendar2	6	Framework
Core::Time::Calendar3	6	Framework
Core::Time::Calendar4	6	Framework

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.

3. Backing Up the System

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

3.1. 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 - backup script  
Copyright (C) 2001-2014 OTRS AG, http://otrs.com/  
usage: backup.pl -d /data_backup_dir/ [-c gzip|bzip2] [-r 30] [-t fullbackup|nofullbackup|  
dbonly]  
linux:/opt/otrs/scripts#
```

Skript: Aufruf der Hilfe des Backup-Mechanismus.

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#
```

Abb.: Erstellen einer Datensicherung.

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#
```

Skript: Prüfung der gesicherten Dateien.

3.2. 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 - restore script  
Copyright (C) 2001-2014 OTRS AG, http://otrs.com/  
usage: restore.pl -b /data_backup/<TIME>/ -d /opt/otrs/  
linux:/opt/otrs/scripts#
```

Skript: Aufruf der Hilfe des Restore-Mechanismus.

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#
```

Skript: Wiederherstellung des Systems.

4. E-Mail-Einstellungen

4.1. Sending/Receiving Emails

4.1.1. Sending Emails

4.1.1.1. Via Sendmail (Default)

OTRS can send out emails via [Sendmail](#), [Postfix](#), [Qmail](#) or [Exim](#). The default configuration is to use Sendmail and should work out-of-the-box.

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

4.1.1.2. Via SMTP Server or Smarthost

OTRS kann E-Mails via SMTP ([Simple Mail Transfer Protocol / RFC 821](#)) oder Secure SMTP versenden.

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

4.1.2. Receiving Emails

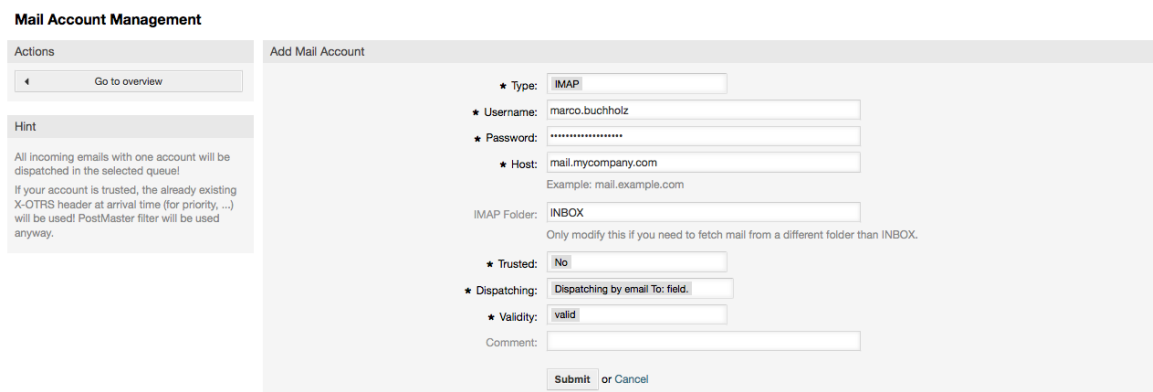
4.1.2.1. Mail Accounts Configured via the OTRS GUI

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

Configure your mail accounts via the "PostMaster Mail Accounts" link on the Admin page.

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

Abbildung 4.66. Einen E-Mail-Account hinzufügen



Mail Account Management

Actions

◀ Go to overview

Hint

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

Add Mail Account

★ Type:

★ Username:

★ Password:

★ Host:
Example: mail.example.com

IMAP Folder:
Only modify this if you need to fetch mail from a different folder than INBOX.

★ Trusted:

★ Dispatching:

★ Validity:

Comment:

or

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

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

All data for the mail accounts are saved in the OTRS database. The `bin/otrs.Console.pl Maint::PostMaster::MailAccountFetch` command uses the settings in the database and fetches the mail. You can execute it manually to check if all your mail settings are working properly.

On a normal installation, the mail will be fetched every 10 minutes by the OTRS Daemon.

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.

4.1.2.2. Via Command Line Program and Procmail (`otrs.Console.pl Maint::PostMaster::Read`)

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

To test `bin/otrs.Console.pl Maint::PostMaster::Read` without an MDA, execute the command of the following script.

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

Skript: PostMaster-Test ohne MDA.

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

Beispiel 4.2. Routing via Procmail Using `otrs.Console.pl`

In order to route mails in a specific queue using `otrs.Console.pl` use the following example.

```
| $SYS_HOME/bin/otrs.Console.pl Maint::PostMaster::Read --target-queue=QUEUE_NAME
```

When sorting to a subqueue, you must separate the parent and child queue with a `::`.

```
| $SYS_HOME/bin/otrs.Console.pl Maint::PostMaster::Read --target-queue=QUEUE_NAME::SUBQUEUE
```

Procmail ist in der Linux-Umgebung ein sehr bekannter E-Mail-Filter, der höchstwahrscheinlich auf Ihrem System installiert sein wird. Falls nicht, erhalten Sie auf der [procmail Homepage](#) 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.Console.pl Maint::PostMaster::Read
```

Skript: Konfiguration von procmail für OTRS.

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

4.1.2.3. Fetching emails via POP3 or IMAP and fetchmail for `otrs.Console.pl Maint::PostMaster::Read`

In order to get email from your mail server, via a POP3 or IMAP mailbox, to the OTRS machine/local OTRS account and to procmail, use [fetchmail](#).

Anmerkung

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

You can use the `.fetchmailrc.dist` in the home directory of OTRS and copy it to `.fetchmailrc`. Modify/change it for your needs (see the Example below).

Beispiel 4.3. `.fetchmailrc`

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

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

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

4.1.2.4. Filtering/Dispatching by OTRS/PostMaster Modules (for More Complex Dispatching)

If you use the `bin/otrs.Console.pl Maint::PostMaster::Read` or `bin/otrs.Console.pl Maint::PostMaster::MailAccountFetch` method, you can insert or modify X-OTRS header entries with the PostMaster filter modules. With the X-OTRS headers, the ticket system can execute some actions on incoming mails, sort them into a specific queue, change the priority or change the customer ID, for example. More infor-

mation about the X-OTRS headers are available in the section about adding mail accounts from the OTRS Admin page.

Es gibt verschiedene Standard-Filtermodule:

Anmerkung

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

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

Beispiel 4.4. Example jobs for the filter module `Kernel::System::PostMaster::Filter::Match`

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

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

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

Beispiel 4.5. 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',
  },
};
```

`Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition` is a default module that adds the possibility to parse external identifiers, in the email subject, the body or both using regular expressions. It then stores this value in a defined dynamic

field. When an email comes in, OTRS will first search for an external identifier and when it finds one, query OTRS on the pre-defined dynamic field. If it finds an existing ticket, it will update this ticket, otherwise it will create a new ticket with the external reference number in the separate field.

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

Beispiel 4.6. Example job for the filter module Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition

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

Konfigurationsoptionen

- FromAddressRegExp

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

- NumberRegExp

This is a mandatory setting. This setting contains the regular expression OTRS will use to extract the ticket number out of the subject and/or ticket body. The default regular expression will match occurrences of for example 'Incident-12354' and will put the part between parentheses in the dynamic field field, in this case '12354'.

- SearchInSubject

Wenn diese Variable auf '1' gesetzt ist, wird der E-Mailbetreff nach einer Ticketnummer durchsucht.

- SearchInBody

Wenn diese Variable auf '1' gesetzt ist, wird der E-Mailbody nach einer Ticketnummer durchsucht.

- TicketStateTypes

This is an optional setting. If given, it will search OTRS only for open external tickets of given state types. The state types are separated with semicolons.

- DynamicField

This is a required setting. It defines the dynamic field that is used to store the external number (the field name must exist in the system and has to be valid).

- SenderType

Definiert den Absender Typ welcher bei der Ticket erstellung verwendet wird.

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

4.1.2.5. Troubleshooting Email Filtering

This section shows some common issues and things to consider when troubleshooting Postmaster filters.

- The filters are worked in order of their alphabetically sorted filter names. The last filter wins for a certain field to be set, when the criteria match twice.
- "Stop After Match" can prevent a second match.
- Make sure the regular expression is valid.
- Headers can be set as to control OTRS, but are not written in the mail itself.
- When matching one From, CC, TO, use EMAILADDRESS: <your@address>
- The Mailbox must be trusted.
- The match criteria are AND conditions.
- Ticket properties can not be matched by the postmaster filter.

4.2. Secure Email with 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. Um GnuPG für OTRS einzurichten, 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:
```


ständigen bzw. zum Ende hin ist an der mit (passphrase) gekennzeichneten Stelle die Passphrase für den zu generierenden Schlüssel einzugeben.

3. In der Oberfläche für die PGPEinstellungen kann PGP für OTRS aktiviert werden (erste Option). Ebenfalls sollte überprüft werden, ob der voreingestellte Pfad zum gpg Programm der tatsächlichen Installation entspricht.

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

Über die letzte Option (PGP::Key::Password) können die Schlüssel-Werte-Paare für die IDs und Passphrasen der eigenen PGP-Schlüssel dem Ticket System bekannt gemacht werden. Noch einmal genauer: da andere Kommunikationspartner an das Ticket-System E-Mails mit dessen öffentlichen Schlüssel verschlüsselt schicken, kann OTRS mit den in dieser Option angegebenen privaten Schlüsseln 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 gpg key for ticket system with
address support@example.com) <support@example.com>
sub   2048g/52B97069 2006-02-03

linux:~$
```

Skript: Die ID des eigenen privaten Schlüssels ermitteln.

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

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

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

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

Die zweite Möglichkeit bietet das System in den PGP-Einstellungen innerhalb des Admin-Bereiches. In dieser Konfigurationsmaske ist im rechten Teil eine Liste der bereits dem System zur Verfügung stehenden öffentlichen Schlüssel zu sehen. Im Regelfall sollte nach der obigen Einrichtung hier schon der öffentliche Schlüssel des Ticket-Sys-

tems 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.

4.3. Secure Email with S/MIME

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

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

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

Skript: Konfiguration von S/MIME.

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

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

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

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

Making CA certificate ...
Generating a 1024 bit RSA private key
...+++++
.....+++++
writing new private key to './demoCA/private/cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
```

```

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

```

Skript: Anlegen einer Certification Authority für SSL.

2. Erzeugen der Zertifikatsanfrage.

```

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

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

```

Skript: Erstellen einer Zertifikatsanfrage.

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

```

otrs@linux:~/tmp> CA.pl -signreq
Using configuration from /etc/ssl/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number:
    fd:85:f6:9f:14:07:16:c8
  Validity
    Not Before: Jan  8 17:04:37 2006 GMT
    Not After : Jan  8 17:04:37 2007 GMT
  Subject:
    countryName           = DE
    stateOrProvinceName   = OTRS-state
    localityName          = OTRS-town
    organizationName      = Your Company
    commonName            = OTRS administrator
    emailAddress          = otrs@your-domain.tld
  X509v3 extensions:
    X509v3 Basic Constraints:
      CA:FALSE
    Netscape Comment:
      OpenSSL Generated Certificate
    X509v3 Subject Key Identifier:
      01:D9:1E:58:C0:6D:BF:27:ED:37:34:14:D6:04:AC:C4:64:98:7A:22
    X509v3 Authority Key Identifier:
      keyid:10:4D:8D:4C:93:FD:2C:AA:9A:B3:26:80:6B:F5:D5:31:E2:8E:DB:A8
      DirName:/C=DE/ST=OTRS-state/L=OTRS-town/O=Your Company/
      CN=OTRS admin/emailAddress=otrs@your-domain.tld
      serial:FD:85:F6:9F:14:07:16:C7

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

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

```

Skript: Signieren der Zertifikatsanfrage.

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

```

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

```

Skript: Neues Zertifikat erstellen.

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

Die OTRS-seitige Einrichtung erfolgt aus dem Admin-Bereich, Block System über den Punkt SMIME. Falls die generelle S/MIME-Unterstützung in OTRS noch nicht aktiviert wur-

de, weist die Maske den OTRS-Administrator beim Aufruf darauf hin und bietet einen bequemen Link zur Einrichtung.

Über die SysConfig-Gruppe "Crypt::SMIME", kann die generelle S/MIME-Unterstützung eingeschaltet und konfiguriert werden.

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.

5. Using External backends

5.1. Customer Data

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 folgenden SysConfig-Parameter, wenn Sie die Kundeninformationen im Agenten-Interface anzeigen wollen.

```
# 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;
```

Skript: SysConfig Konfigurationsparameter.

5.2. Customer User 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.

5.2.1. Datenbank (Standard)

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

Beispiel 4.7. 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',

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

# customer #
CustomerID => 'customer_id',
CustomerValid => 'valid_id',
CustomerUserListFields => ['first_name', 'last_name', 'email'],
CustomerUserSearchFields => ['login', 'last_name', 'customer_id'],
CustomerUserSearchPrefix => '',
CustomerUserSearchSuffix => '*',
CustomerUserSearchListLimit => 250,
CustomerUserPostMasterSearchFields => ['email'],
CustomerUserNameFields => ['title', 'first_name', 'last_name'],
CustomerUserEmailUniqCheck => 1,
# # show not own tickets in customer panel, CompanyTickets
# CustomerUserExcludePrimaryCustomerID => 0,
# # generate auto logins
# AutoLoginCreation => 0,
# AutoLoginCreationPrefix => 'auto',
# # admin can change customer preferences
# AdminSetPreferences => 1,
# # cache time to live in sec. - cache any database queries
# CacheTTL => 0,
# # just a read only source
# ReadOnly => 1,
  Map => [
    # note: Login, Email and CustomerID needed!
    # var, frontend, storage, shown (1=always,2=lite), required, storage-type, http-
link, readonly, http-link-target
    [ 'UserTitle', 'Title', 'title', 1, 0, 'var', '', 0 ],
    [ 'UserFirstname', 'Firstname', 'first_name', 1, 1, 'var', '', 0 ],
    [ 'UserLastname', 'Lastname', 'last_name', 1, 1, 'var', '', 0 ],
    [ 'UserLogin', 'Username', 'login', 1, 1, 'var', '', 0 ],
    [ 'UserPassword', 'Password', 'pw', 0, 0, 'var', '', 0 ],
    [ 'UserEmail', 'Email', 'email', 1, 1, 'var', '', 0 ],
# [ 'UserEmail', 'Email', 'email', 1, 1, 'var', "[% Env('CGIHandle')
%]?Action=AgentTicketCompose&ResponseID=1&TicketID=[% Data.TicketID %]&ArticleID=[%
Data.ArticleID %]", 0 ],
    [ 'UserCustomerID', 'CustomerID', 'customer_id', 0, 1, 'var', '', 0 ],
# [ 'UserCustomerIDs', 'CustomerIDs', 'customer_ids', 1, 0, 'var', '', 0 ],
    [ 'UserPhone', 'Phone', 'phone', 1, 0, 'var', '', 0 ],
    [ 'UserFax', 'Fax', 'fax', 1, 0, 'var', '', 0 ],
    [ 'UserMobile', 'Mobile', 'mobile', 1, 0, 'var', '', 0 ],
    [ 'UserStreet', 'Street', 'street', 1, 0, 'var', '', 0 ],
    [ 'UserZip', 'Zip', 'zip', 1, 0, 'var', '', 0 ],
    [ 'UserCity', 'City', 'city', 1, 0, 'var', '', 0 ],
    [ 'UserCountry', 'Country', 'country', 1, 0, 'var', '', 0 ],
    [ 'UserComment', 'Comment', 'comments', 1, 0, 'var', '', 0 ],
    [ 'ValidID', 'Valid', 'valid_id', 0, 1, 'int', '', 0 ],
  ]
}

```

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

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

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

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

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

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

mysql> quit
Bye
linux:~#
```

Skript: Ein Feld zur Tabelle customer_user hinzufügen.

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

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

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

Es ist auch möglich, diese Kundeninformationen über den Kundenlink im Agent-Interface zu bearbeiten.

5.2.1.1. Customer with Multiple IDs (Company Tickets)

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

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

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

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

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

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

mysql> quit
Bye
linux:~#
```

Skript: Hinzufügen eines Feldes customer_ids zur Tabelle customer_user.

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

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

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

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

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

Beispiel 4.8. Using Company Tickets with a DB 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.

5.2.2. LDAP

If you have an LDAP directory with your customer data, you can use it as the customer backend with OTRS, as shown in Example below.

Beispiel 4.9. 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 the charset of your ldap server is iso-8859-1, use this:
#     SourceCharset => '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 an ldap filter for valid users (expert setting)
# CustomerUserValidFilter => '!(description=locked)',
# administrator can't change customer preferences
AdminSetPreferences => 0,
# # cache time to live in sec. - cache any database queries
# CacheTTL => 0,
Map => [
    # note: Login, Email and CustomerID are mandatory!
    # var, frontend, storage, shown (1=always,2=lite), required, storage-type, http-
link, readonly
    [ 'UserTitle',      'Title',      'title',          1, 0, 'var', '', 0 ],
    [ 'UserFirstname', 'Firstname', 'givenname',     1, 1, 'var', '', 0 ],
    [ 'UserLastname',  'Lastname',  'sn',            1, 1, 'var', '', 0 ],
    [ 'UserLogin',     'Username',  'uid',           1, 1, 'var', '', 0 ],
    [ 'UserEmail',     'Email',     'mail',          1, 1, 'var', '', 0 ],
    [ 'UserCustomerID', 'CustomerID', 'mail',          0, 1, 'var', '', 0 ],
#     [ 'UserCustomerIDs', 'CustomerIDs', 'second_customer_ids', 1, 0, 'var', '', 0 ],
    [ 'UserPhone',     'Phone',     'telephonenumber', 1, 0, 'var', '', 0 ],
    [ 'UserAddress',   'Address',   'postaladdress',  1, 0, 'var', '', 0 ],
    [ 'UserComment',   'Comment',   'description',    1, 0, 'var', '', 0 ],
],
};

```

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

```

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

```

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

5.2.2.1. Customer with Multiple IDs (Company Tickets)

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

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

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

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

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

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

Beispiel 4.10. 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.

5.2.3. Using More than One Customer Backend with OTRS

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

Beispiel 4.11. 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 the charset of your ldap server is iso-8859-1, use this:
        SourceCharset => '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.

5.2.4. Storing CustomerUser Data in Dynamic Fields

Manchmal kann es von Vorteil sein Kundendaten direkt in einem dynamischen Feld innerhalb eines Ticket zu speichern, um z.B. spezielle Statistiken zu erzeugen.

The dynamic field values are set when a ticket is created or when the customer of a ticket is changed. The values of the dynamic fields are taken from the customer data. This works for all backends, but is especially useful for LDAP-backends.

To activate this optional feature of OTRS, please activate the settings "Ticket::EventModulePost###930-DynamicFieldFromCustomerUser" and "DynamicFieldFromCustomerUser::Mapping". The latter setting contains the configuration of which CustomerUser field entry should be stored in which ticket dynamic field. The fields must be present in the system and should be enabled for AgentTicketFreeText, so that they can be set manually. They mustn't be enabled for AgentTicketPhone, AgentTicketEmail and AgentTicketCustomer. If they were, they would have precedence over the automatically set values.

5.3. Backends to Authenticate Agents and Customers

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

5.3.1. Authentifizierungs-Backends für Agenten

5.3.1.1. Datenbank (Standard)

The backend to authenticate agents which is used by default is the OTRS database. Agents can be added and edited via the agent management interface in the Admin page (see Example below).

Beispiel 4.12. Agentenauthentifizierung gegen ein Datenbank-Backend

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

5.3.1.2. LDAP

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

Beispiel 4.13. 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.com
# $Self->{'AuthModule::LDAP::UserSuffix'} = '@domain.com';

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

Mit den folgenden Konfigurationsparametern können die Benutzerdaten der Agenten aus dem LDAP in die lokale OTRS Datenbank synchronisiert werden. Dies reduziert die Zugriffe auf ihr LDAP Verzeichnis, entlastet den Server mit den LDAP Daten und beschleunigt die

Anmeldung an OTRS. Die Synchronisierung der Daten findet bei der ersten Anmeldung des Agenten statt, trotz der synchronisierten Daten bleibt ihr LDAP Verzeichnis die letzte Instanz bei der Anmeldung. D.h. wird ein User im LDAP Verzeichnis gelöscht oder deaktiviert, klappt die Anmeldung an OTRS nicht. Ebenfalls müssen die Daten für einen Agenten weiterhin direkt im LDAP Verzeichnis gepflegt werden.

```
# defines AuthSyncBackend (AuthSyncModule) for AuthModule
# if this key exists and is empty, there won't be a sync.
# example values: AuthSyncBackend, AuthSyncBackend2
$self->{'AuthModule::UseSyncBackend'} = 'AuthSyncBackend';

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

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

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

Alternatively, you can use LDAP groups to determine group memberships or roles in OTRS. For more information and examples, see `Kernel/Config/Defaults.pm`. Here is an example for synchronizing from LDAP into OTRS groups.

```
# Attributes needed for group syncs
# (attribute name for group value key)
$self->{'AuthSyncModule::LDAP::AccessAttr'} = 'memberUid';
# (select the attribute for type of group content UID/DN for full ldap name)
# $self->{'AuthSyncModule::LDAP::UserAttr'} = 'UID';
# $self->{'AuthSyncModule::LDAP::UserAttr'} = 'DN';

AuthSyncModule::LDAP::UserSyncGroupsDefinition
# (If "LDAP" was selected for AuthModule and you want to sync LDAP
# groups to otrs groups, define the following.)
$self->{'AuthSyncModule::LDAP::UserSyncGroupsDefinition'} = {
    # your ldap group
    'cn=agent,o=otrs' => {
        # otrs group(s)
        'admin' => {
            # permission
            rw => 1,
            ro => 1,
        },
        'faq' => {
            rw => 0,
            ro => 1,
        },
    },
    'cn=agent2,o=otrs' => {
        'users' => {
            rw => 1,
            ro => 1,
        },
    },
};
```

```

    },
  }
};

```

5.3.1.3. HTTPBasicAuth für Agenten

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

Beispiel 4.14. Agentenauthentifizierung über HTTPBasic

```

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

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

```

5.3.1.4. Radius

The configuration parameters shown in Example below can be used to authenticate agents against a Radius server.

Beispiel 4.15. 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';

```

5.3.2. Authentication Backends for Customers

5.3.2.1. Datenbank (Standard)

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

Beispiel 4.16. Kundenauthentifizierung gegen ein Datenbank-Backend

```

# This is the auth. module against the otrs db
$Self->{'Customer::AuthModule'} = 'Kernel::System::CustomerAuth::DB';
$Self->{'Customer::AuthModule::DB::Table'} = 'customer_user';
$Self->{'Customer::AuthModule::DB::CustomerKey'} = 'login';
$Self->{'Customer::AuthModule::DB::CustomerPassword'} = 'pw';
#$Self->{'Customer::AuthModule::DB::DSN'} =
  "DBI:mysql:database=customerdb;host=customerdbhost";
#$Self->{'Customer::AuthModule::DB::User'} = "some_user";
#$Self->{'Customer::AuthModule::DB::Password'} = "some_password";

```

5.3.2.2. LDAP

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

Beispiel 4.17. 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.com
# $self->{'Customer::AuthModule::LDAP::UserSuffix'} = '@domain.com';

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

5.3.2.3. HTTPBasicAuth for Customers

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

Beispiel 4.18. Kundenauthentifizierung über HTTPBasic

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

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

5.3.2.4. Radius

The settings shown in Example below can be used to authenticate your customers against a Radius server.

Beispiel 4.19. 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';
```

5.4. 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.

5.4.1. Customizing the Web Interface

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

```
[...]
<div class="NewLine">
  <label for="Room">[% Translate("Room{CustomerUser}") | html %]</label>
  <input title="[% Translate("Room Number") | html %]" name="Room" type="text"
  id="UserRoom" maxlength="50" />
</div>
[...]
```

Skript: Anzeigen eines neuen Feldes in der Weboberfläche.

5.4.2. Customer 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.',
    },
},
};

```

Skript: Anpassung des MAP-Arrays.

5.4.3. 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:~#
```

Skript: Hinzufügen einer neuen Spalte zur Tabelle customer_user.

Now the new field for the room should be displayed in the Customer Information panel if filled, and in the Customer User administration screens. Also, new customers should have to insert their room number if they register a new account.

6. Ticket-Einstellungen

6.1. Ticket Status

6.1.1. 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'.

6.1.1.1. Neu

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

6.1.1.2. Öffnen

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

6.1.1.3. 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.

6.1.1.4. 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.

6.1.1.5. 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.

6.1.1.6. Zusammengefügt

Status von Tickets, die mit anderen Tickets zusammengefügt wurden.

6.1.1.7. Erfolgreich geschlossen

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

6.1.1.8. Erfolglos geschlossen

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

6.1.2. Anpassen von Ticket-Status

Every state has a name (state-name) and a type (state-type). Click on the States link on the Admin page and press the button "Add state" to create a new state. You can freely choose the name of a new state. The state types can not be changed via the web interface. The database has to be directly modified if you want to add new types or change existing names. The default state types should typically not be modified as this can yield unpredictable results. For instance, escalation calculations and the unlock feature are based on specific state types.

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

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

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

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

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

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

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

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

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

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

Skript: Anpassung der Statustypen.

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

Ticket -> Frontend::Agent::Ticket::ViewPhoneNew > AgentTicketPhone###StateDefault - um den vorgegebenen nächsten Status für neue Telefon-Tickets zu definieren.

Ticket -> Frontend::Agent::Ticket::ViewPhoneNew > AgentTicketPhone###StateType - um die verfügbaren nächsten Status für neue Telefon-Tickets zu definieren.

Ticket -> Frontend::Agent::Ticket::ViewEmailNew > AgentTicketEmail###StateDefault - um den standardmäßigen 'Nächsten Status des Tickets' für E-Mailtickets zu definieren.

Ticket -> Frontend::Agent::Ticket::ViewEmailNew > AgentTicketEmail###StateType - um die verfügbaren 'Nächsten Status' für neue E-Mailtickets zu definieren.

Ticket -> Frontend::Agent::Ticket::ViewPhoneOutbound > AgentTicketPhoneOutbound###State - um den standardmäßigen 'Nächsten Status des Tickets' für neue Artikel in Telefontickets zu definieren.

Ticket -> Frontend::Agent::Ticket::ViewPhoneOutbound > AgentTicketPhoneOutbound###StateType - um die verfügbaren 'Nächsten Status' für neue Artikel in Telefontickets zu definieren.

Ticket -> Frontend::Agent::Ticket::ViewMove > AgentTicketMove###State - um den Standard-Status für Tickets zu definieren, wenn ein Ticket verschoben wird.

Ticket -> Frontend::Agent::Ticket::ViewMove > AgentTicketMove###StateType - um die verfügbaren nächsten Status eines Tickets zu definieren, wenn ein Ticket verschoben wird.

Ticket -> Frontend::Agent::Ticket::ViewBounce > StateDefault - um den Standard-Status für Tickets zu definieren, wenn ein Ticket weitergeleitet wird.

Ticket -> Frontend::Agent::Ticket::ViewBounce > StateType - um die verfügbaren nächsten Status eines Tickets zu definieren, wenn ein Ticket weitergeleitet wird.

Ticket -> Frontend::Agent::Ticket::ViewBulk > StateDefault - um den Standard-Status für Tickets zu definieren, wenn eine Sammelaktion durchgeführt wird.

Ticket -> Frontend::Agent::Ticket::ViewBulk > StateType - um die verfügbaren nächsten Status eines Tickets zu definieren, wenn eine Sammelaktion durchgeführt wird.

Ticket -> Frontend::Agent::Ticket::ViewClose > StateDefault - um den Standard-Status für Tickets zu definieren, wenn ein Ticket geschlossen werden soll.

Ticket -> Frontend::Agent::Ticket::ViewClose > StateType - um die verfügbaren nächsten Status eines Tickets zu definieren, wenn ein Ticket geschlossen werden soll.

Ticket -> Frontend::Agent::Ticket::ViewCompose > StateDefault - um den Standard-Status für Tickets zu definieren, wenn auf ein Ticket geantwortet (AgentTicketCompose) werden soll.

Ticket -> Frontend::Agent::Ticket::ViewCompose > StateType - um die verfügbaren nächsten Status eines Tickets zu definieren, wenn auf ein Ticket geantwortet (AgentTicketCompose) werden soll.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateDefault - um den Standard-Status für Tickets zu definieren, wenn ein Ticket weitergeleitet werden soll.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateType - um die verfügbaren nächsten Status eines Tickets zu definieren, wenn ein Ticket weitergeleitet werden soll.

Ticket -> Frontend::Agent::Ticket::ViewForward > StateDefault - um den Standard-Status für Tickets in der Oberfläche 'Freie Felder' zu definieren,

Ticket -> Frontend::Agent::Ticket::ViewForward > StateType - um die verfügbaren nächsten Status eines Tickets in der Oberfläche 'Freie Felder' zu definieren.

Ticket -> Core::PostMaster > PostmasterDefaultState - um den Status von Tickets zu definieren, die aus E-Mails erstellt wurden.

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

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

Ticket -> Core::Ticket > ViewableStateType - um die Statustypen festzulegen, die in verschiedenen Bereichen des Systems angezeigt werden, z. B. in der Queue-Ansicht.

Ticket -> Core::Ticket > UnlockStateType - um die verfügbaren Statustypen für entspernte Tickets zu definieren.

Ticket -> Core::Ticket > PendingReminderStateType - um den Statustyp für Erinnerungstickets festzulegen.

Ticket -> Core::Ticket > PendingAutoStateType - um den Statustyp für Tickets mit "Warten Auto" festzulegen.

Ticket -> Core::Ticket > StateAfterPending - um den Status zu definieren, auf welchen Tickets, deren "Warten Auto"-Timer des eingestellten Status abgelaufen ist, gesetzt werden.

6.2. Ticket Priorities

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 Standard-Priorität für alle eingehenden E-Mails.

- Ticket:Frontend::Agent:Ticket::ViewPhoneNew:Priority - definiert die Standard-Priorität in der Oberfläche 'Neues Telefonticket' für Agenten.
- Ticket:Frontend::Agent:Ticket::ViewEmailNew:Priority - definiert die Standard-Priorität in der Oberfläche 'Neues E-Mailticket' für Agenten.
- Ticket:Frontend::Customer:Ticket::ViewNew:PriorityDefault - definiert die Standard-Priorität in der Oberfläche 'Neues Ticket' im Kunden-Interface.

6.3. 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 auch von jemand anderem als dem Besitzer 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.

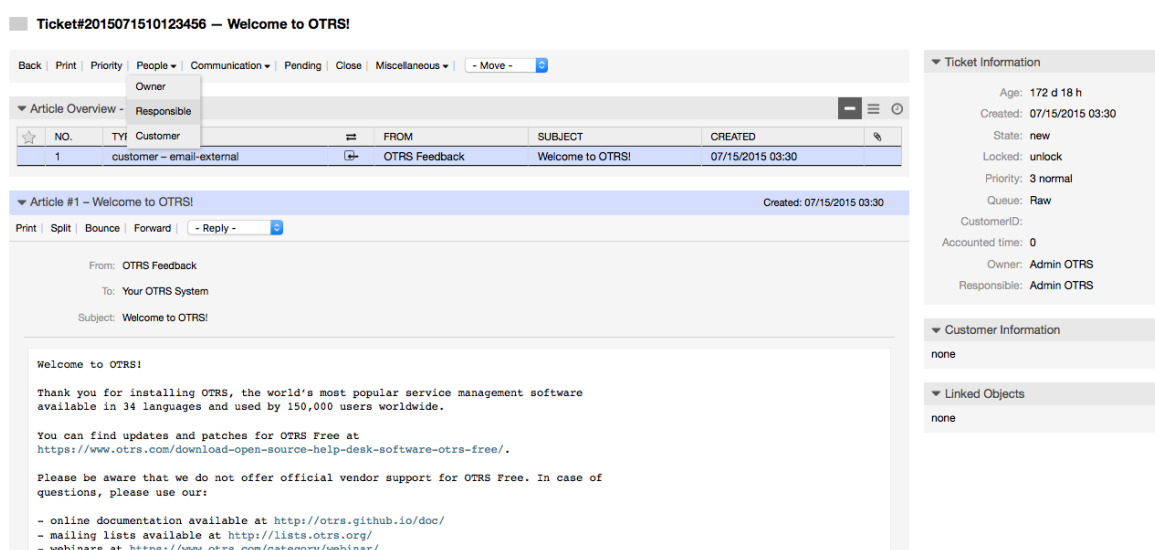
6.3.1. Ticket-Verantwortlicher

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

Mit dem Konfigurationsparameter Ticket::Responsible kann die Funktion TicketResponsible aktiviert werden. Dadurch werden 3 neue Icons in ihrer Toolbar erscheinen.

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

Abbildung 4.67. Changing the Responsibility of a ticket in its zoomed view



The screenshot shows the OTRS ticket interface for ticket #2015071510123456. The main content area displays the ticket details, including the subject 'Welcome to OTRS!' and the sender 'OTRS Feedback'. A dropdown menu is open over the 'Responsible' field, showing the current responsible agent as 'Admin OTRS'. The right sidebar contains 'Ticket Information' (Age: 172 d 18 h, State: new, Priority: 3 normal) and 'Customer Information' (none).

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

Abbildung 4.68. Pop-up dialog to change a ticket's responsibility

Change Responsible of Ticket#2015071510123456 – Welcome to OTRS!

All fields marked with an asterisk (*) are mandatory.

Cancel & close

▼ Ticket Settings

*Title:

New Responsible:

▶ Add Article

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.

6.3.2. 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.

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

Abbildung 4.69. Subscribing to watching a ticket in its zoomed view

Ticket#2015071510123456 – Welcome to OTRS!

Back | Print | Priority | People | Communication | Pending | Watch | Close | Miscellaneous | - Move -

▼ Article Overview - 1 Article(s)

NO.	TYPE	FROM	SUBJECT	CREATED
1	customer – email-external	OTRS Feedback	Welcome to OTRS!	07/15/2015 03:30

▼ Article #1 – Welcome to OTRS! Created: 07/15/2015 03:30

Print | Split | Bounce | Forward | - Reply -

From: OTRS Feedback
To: Your OTRS System
Subject: Welcome to OTRS!

Welcome to OTRS!

Thank you for installing OTRS, the world's most popular service management software available in 34 languages and used by 150,000 users worldwide.

▼ Ticket Information

Age: 172 d 18 h
Created: 07/15/2015 03:30
State: new
Locked: unlock
Priority: 3 normal
Queue: Raw
CustomerID:
Accounted time: 0
Owner: Admin OTRS
Responsible: Admin OTRS

▼ Customer Information

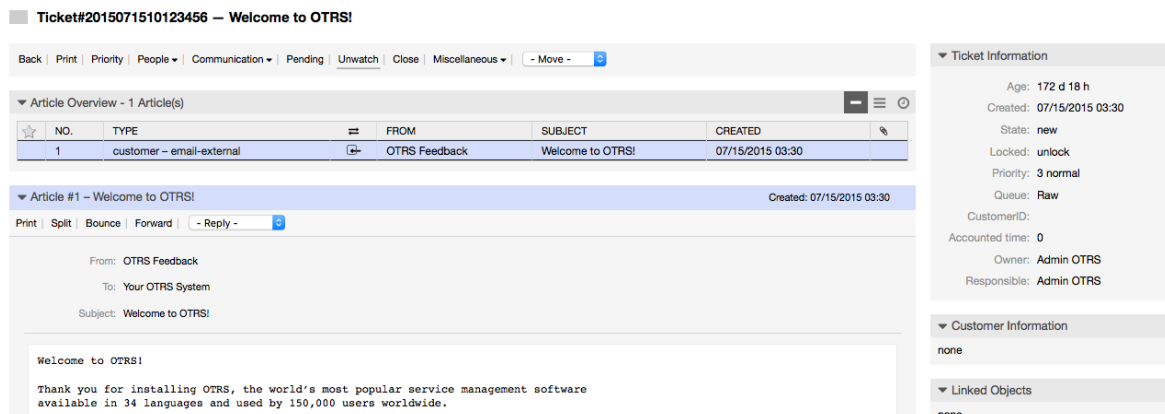
none

▼ Linked Objects

none

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

Abbildung 4.70. Unsubscribing from watching a ticket in its zoomed view



The list of all watched tickets can be accessed through the Watched view of the OTRS agent interface (see figure below), as soon as the ticket watcher feature gets activated.

Abbildung 4.71. Watched tickets view



TICKET#	AGE	FROM / SUBJECT	STATE	LOCK	QUEUE	OWNER	CUSTOMERID
2015071510123456	172 d 18 h	OTRS Feedback Welcome to OTRS!	new	unlock	Raw	Admin OTRS	

7. Date and Time Related Functions

7.1. Arbeitszeiten, Feiertage und Zeitzonen für das System festlegen

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

7.1.1. Time zones

OTRS needs to know which time zone should be used to store date and time related data in the database. You can set this in Framework > Core::Time::OTRSTimeZone. The default is UTC and if you set up a new OTRS it is strongly recommended to leave it at this. If you updated from an OTRS older than version 6, you must ensure that OTRSTimeZone is being set to a time zone that matches your previous setup. Otherwise new data will be stored with a different time zone than your existing data. Once you decided on a time zone and data was stored (tickets, etc.), you shouldn't change the time zone anymore because otherwise you would end up with data stored in different time zones.

You can set a default time zone for new agents and customer users via Framework > Core::Time::UserDefaultTimeZone. This time zone will be used for all users that don't have selected a time zone in their preferences.

7.1.2. Arbeitszeiten

Set up the working hours for your system in SysConfig Framework > Core::Time::TimeWorkingHours, or for your specific calendar in the calendar's configuration. OTRS can handle a granularity of one hour. Checking the marks in the boxes 8, 9, 10 ... 17 corresponds with business hours of 8:00 AM - 6:00 PM.

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

7.1.3. Fixed Date Holidays

Feiertage deren Datum jedes Jahr gleich ist, wie Neujahr oder der vierte Juli, können in TimeVacationDays oder im entsprechenden Abschnitt in den Kalendern 1-9 konfiguriert werden.

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

Anmerkung

In OTRS sind die *deutschen* Feiertage voreingestellt.

7.1.4. Floating Holidays

Feiertage, wie Ostern, für die sich jährlich das Datum ändert, können in TimeVacationDaysOneTime konfiguriert 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.

7.2. 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.

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

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

Tickets werden entsperrt wenn folgende Bedingungen zutreffen:

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

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

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

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

8. Customizing the PDF Output

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

If you use the Print action from anywhere within the OTRS interface, it will generate a formatted PDF file.

You can adjust the look of the files generated by OTRS by creating your own logo and adding it to PDF::LogoFile. You can use PDF::PageSize to define the standard page size of the generated PDF file (DIN-A4 or Letter), and also PDF::MaxPage to specify the maximum number of pages for a PDF file, which is useful if a user generates a huge output file by mistake.

9. Statistiken

The OTRS statistics module holds features to track operational statistics and generates custom reports associated with OTRS usage. The OTRS system uses the term "statistic" generically to refer to a single report presenting various indicators.

Anmerkung

For **OTRS Business Solution™** customers, there is also a reports generator available. Here "report" refers to a collection of several statistics in one PDF document that can be easily configured and automatically generated and distributed. Please find more details in the **OTRS Business Solution™** manual.

Proper configuration of the OTRS statistics module is associated with a multitude of requirements and considerations. These include the various OTRS modules to be evaluated, user permission settings, indicators to be calculated and their complexity levels, ease of configuration of the statistics module, speed and efficiency of calculations, and support of a rich set of output variants.

Statistical elements, i.e. files which supplement the functionality of the statistics module for specific requirements, can be integrated for calculating complex statistics.

9.1. Statistics Configuration and Usage

When signed on as an agent, the statistics module can be opened by selecting "Reports" and then "Statistics" in the main menu.

9.1.1. Übersicht

Selecting the "Statistics" link in the navigation bar, and then the submenu link "Overview", calls up the Overview screen. The Overview screen presents a list of all pre-configured reports the agent can use (see figure below).

Abbildung 4.72. Overview of the standard statistics.

Statistics » Overview

STAT#	TITLE	OBJECT	EXPORT	DELETE	RUN
10001	List of open tickets, sorted by time left until escalation deadline expires	Ticketlist			Run now
10002	List of open tickets, sorted by time left until response deadline expires	Ticketlist			Run now
10003	List of open tickets, sorted by time left until solution deadline expires	Ticketlist			Run now
10004	List of the most time-consuming tickets	Ticketlist			Run now
10005	List of tickets closed last month	Ticketlist			Run now
10006	List of tickets closed, sorted by response time.	Ticketlist			Run now
10007	List of tickets closed, sorted by solution time	Ticketlist			Run now
10008	List of tickets created last month	Ticketlist			Run now
10009	New Tickets	TicketAccumulation			Run now
10010	Changes of status in a monthly overview	StateAction			Run now
10011	Overview about all tickets in the system	TicketAccumulation			Run now

When the statistics module is installed, it comes preloaded with a few sample statistics imported into the system. These are shown as a list on the overview page. If the overview list extends to more than a single page, the agent can browse through the different pages. The list of statistics can be sorted as desired, by clicking the desired column header in the list. To generate a particular statistic, click on the statistic's "Run now" link.

9.1.2. Generation

The view user interface provides the stat's configuration settings (see figure below).

Abbildung 4.73. Viewing a specific statistic.

Statistics » View Stat#10001 — List of open tickets, sorted by time left until escalation deadline expires

Actions	Settings
<input type="button" value="Go to overview"/> <input type="button" value="Edit"/>	Object: Ticketlist Description: List of open tickets, sorted by time left until escalation deadline expires. NOTE: Please check the output and configuration of the statistics carefully to make sure that it produces the results you expect. If necessary, change the configuration before using the statistics in a production environment. Format: <input type="text" value="CSV"/> X-axis Attributes to be printed: Number, Ticket#, Age, Title, Created, Changed, Close Time, Queue, State, Priority, Customer User, CustomerID, Service... Y-axis Order by: EscalationTimeWorkingTime Sort sequence: ascending Filter State: new, open, pending auto close+, pending auto close-, pending reminder <input type="button" value="Run now"/> or Cancel
Statistic Information Created: 12/14/2015 09:32:45 Created by: test1450081960626627533 test1450081960626627533 Changed: 12/14/2015 09:32:45 Changed by: test1450081960626627533 test1450081960626627533 Sum rows: No Sum columns: No Show as: No dashboard widget: Cache: No Validity: valid	

Configuration settings for a particular statistic can be set within the range of options in the View screen. Either the statistic creator or any others with the appropriate permissions can make the settings.

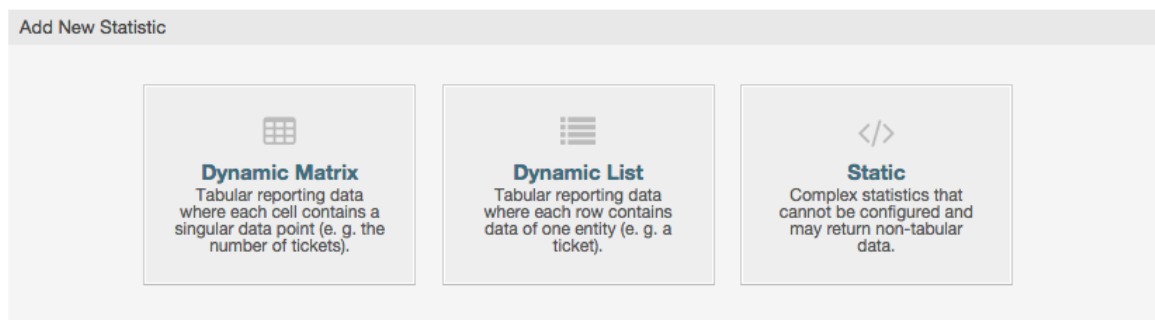
Pressing the "Start" button (at the bottom of the screen) is the last step to generate the statistic.

9.1.3. Konfiguration

Agents with write rights can edit an existing report configuration by calling up the edit user interface of the statistics module. Alternately, they may create a new report.

There are four possible steps in the configuration of a statistic: the general specification data, configuring the x-axis, y-axis and possible data filters for the reported data (or restrictions). Let's create a new statistic as an example by clicking the "Add" button in the overview screen. Our goal will be to get an overview of how many tickets with very high priority are in every queue (x-axis) and state (y-axis).

Abbildung 4.74. Adding a new statistic, first step.



At the beginning we have to select the type of statistic we want to add. Three types are available:

Dynamic Matrix Statistics

This type of statistics will generate a matrix of computed values (e.g. new tickets per day of month and queue). All value cells in the matrix have the same type (number, average time, etc.). Values are computed from entities in the system (e.g. tickets). Some matrix statistics support a summation column and/or row (only useful for certain data).

Dynamic List Statistics

This kind of statistic will generate a table where every line (not cell) represents an entity in the system (e. g. a ticket). The columns in this row are usually configurable (x-axis, see below) and contain the data of this object (e. g. ticket attributes). All value cells in one column have the same type.

Static Statistics

This kind of statistic is not very much configurable and usually used for very special and/or complex computations.

So let's select "Dynamic Matrix" for our example. Then the "General Specifications" configuration will appear below the statistic type selection.

Abbildung 4.75. Adding a new statistic, second step.

General Specification

★ Title:

★ Description:

★ Object type:

★ Permissions:
You can select one or more groups to define access for different agents.

★ Result formats:

Create summation row:
Generate an additional row containing sums for all data columns.

Create summation column:
Generate an additional column containing sums for all data rows.

Cache results:
Stores statistics result data in a cache to be used in subsequent views with the same configuration.

Validity:
If set to invalid end users can not generate the stat.

Create Statistic

or

After providing a title and description for the new statistic, we have to select the statistics backend that we want to use. This is the actual backend module which is responsible to collect and analyze the data for our statistic. In our case we'll select "TicketAccumulation".

By configuring permission groups, we can facilitate a restriction of the groups (and therefore, agents) who can later view and generate the pre-configured statistics. Thus the various statistics can be allocated to the different departments and work groups who need them. It is possible to allocate one statistic to various groups.

Beispiel 4.20. Default statistics permission group

The "stats" group was selected. The report is viewable for all users having at least ro rights for the "stats" group. This access is available by default.

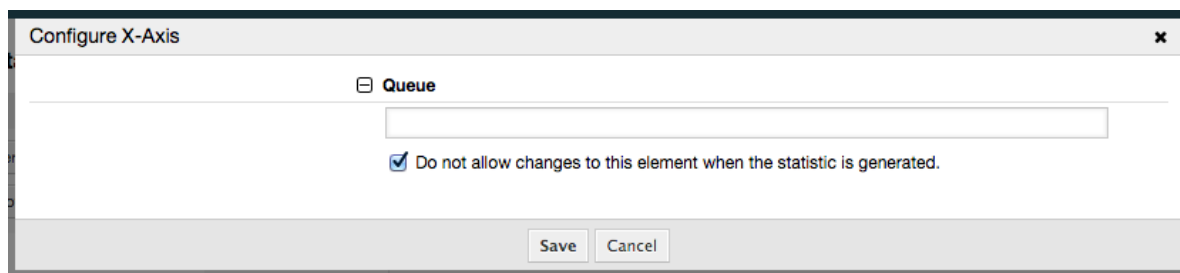
Beispiel 4.21. Customized statistics permission group

A group named "sales" was selected. All users with ro rights for the "sales" group can see the stat in the view mode and generate it. However, the report will not be available for viewing by other users.

Additionally, possible output formats can be selected. Here we can just keep all output formats and choose the one to use when actually generating the statistic. Let's save the statistic now.

The next screen will indicate the next step with a highlighted button: we should configure the x-axis. By clicking the button, a dialog will appear where we can select the element to be used for the x-axis. In our case that will be the queue:

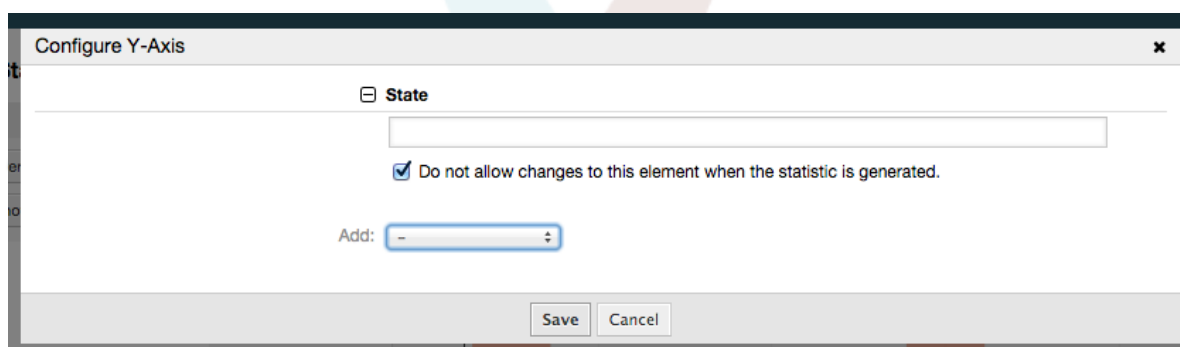
Abbildung 4.76. Configuring the x-axis of a statistic.



We can optionally limit the queues to be shown by selecting some in the queue field. With the checkbox we can control if the agent who generates the statistic can make changes to the queue selection. We'll keep the defaults and press "Save".

Now we can configure the y-axis in the same way: select the state field.

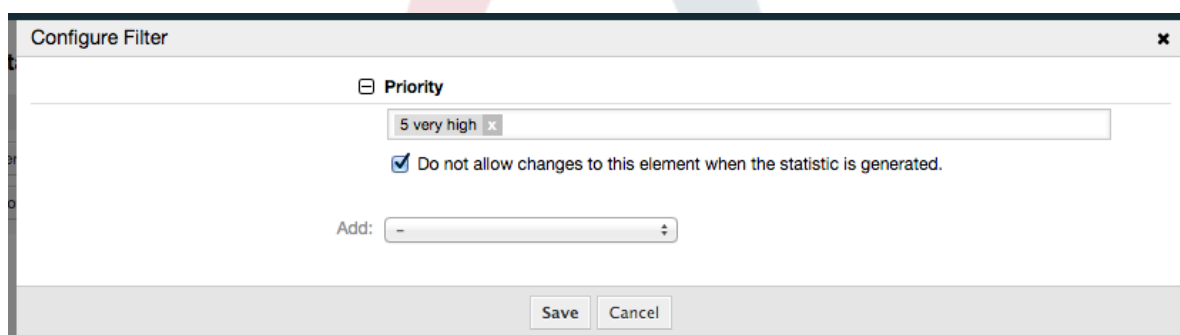
Abbildung 4.77. Configuring the y-axis of a statistic.



Here it is possible to select one element or two. In the first case, every value of the element will be one element on the y-axis. If two elements are selected, their permutations will be the elements on the value series. For example you could select "state" and "priority", and the resulting elements will be "new - 1 very low", "new - 2 low", ... "open - 1 very low" and so on. Let's just use the state and press "Save".

Now in the last step we could add data filters to only report tickets belonging to a certain customer, with certain priorities and so on. We'll add a filter for very high priority tickets:

Abbildung 4.78. Configuring the data filter of a statistic.



Now press "Save" again. The configuration is finished.

You may already have noted that in the configuration dialog there is a preview area where we can check the effect of our configuration settings.

Abbildung 4.79. Configuring the data filter of a statistic.



Anmerkung

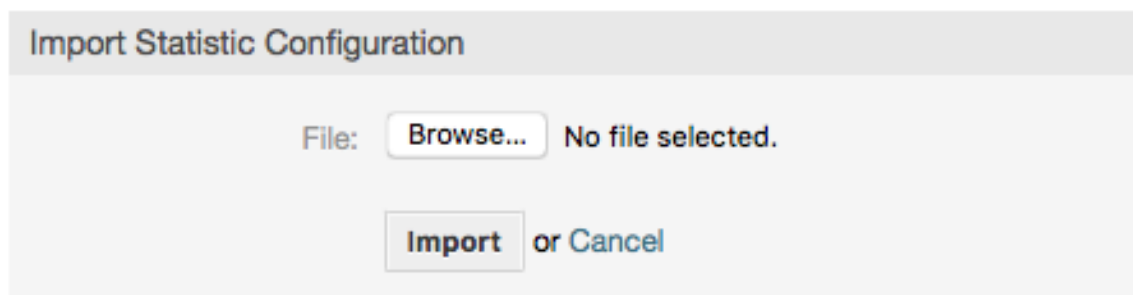
Bitte beachten Sie, dass die Vorschau Zufallsdaten verwendet und keine Datenfilter berücksichtigt.

The statistic is configured. By pressing the "Run now" button we can go to the View screen where the desired output format can be selected and the statistic can be generated in the different formats.

9.1.4. Importieren

The Import user interface can be accessed by pressing the "Import" button on the Overview screen. "rw" permissions for the statistics module are required.

Abbildung 4.80. Statistics import



Facilitates the import of reports, and when combined with the export function of the module, is a very handy functionality. Stats can be created and tested conveniently on test systems, then imported into the production system.

9.2. Statistics System Administration

In diesem Abschnitt werden die Aufgaben und Möglichkeiten des OTRS-Administrators in Bezug auf das Statistikmodul angesprochen.

9.2.1. Rechte, Gruppen und Queues

The default configuration of the module registration gives all agents with "stats" group permissions access to the statistics module.

Zugangsmöglichkeiten aufgrund der Rechte:

- *rw*. Allows configuring statistics.
- *ro*. Permits generating pre-configured statistics.

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.

9.2.2. SysConfig Settings

The SysConfig groups Framework:Core::Stats and Framework:Frontend::Agent::Stats contain all configuration parameters for the basic set-up of the statistics module. Moreover, the configuration parameter `$Self->{'Frontend::Module'}->{'AgentStats'}` controls the arrangement and registration of the modules and icons within the statistics module.

9.2.3. Generating Statistics on the Command Line

Statistics can be generated on the command line with the command `bin/otrs.Console.pl Maint::Stats::Generate`. As an example, see the command line call in the following script.

```
shell> bin/otrs.Console.pl Maint::Stats::Generate --number 10004 --target-directory /tmp
Generating statistic number 10004...
Writing file /tmp/List_of_the_most_time-consuming_tickets_Created_2015-09-08_14-51.csv.
Done.
```

A report from the statistic configuration "Stat#10004" is generated and saved as a CSV file in the /tmp directory.

The generated report can also be sent as an e-mail. More information can be called up with the command in the script below.

```
shell> bin/otrs.Console.pl Maint::Stats::Generate --help
```

It usually does not make sense to generate reports manually via the command line, as the statistics module has a convenient graphical user interface. However, generating reports manually does make sense when combined with a cron job.

Imagine the following scenario: On the first day of every month, the heads of department want to receive a report for the past month. By combining a cron job and command line call the reports can be sent to them automatically by e-mail.

10. Dynamische Felder

10.1. Einführung

Ein dynamisches Feld ist ein ein spezielles Feld in OTRS um die am Ticket oder Artikel gespeicherten Informationen zu erweitern. Diese Felder werden nicht fix im System festgelegt und sind nur in bestimmten Oberflächen sichtbar, sie können verpflichtend sein oder nicht und ihre Darstellung in Oberflächen ist abhängig vom Feldtyp, der während der Erstellung gewählt wurde. Zum Beispiel gibt es Felder, um einen Text, ein Datum, usw. eine Auswahl zu erfassen.

Dynamic fields are the evolution of TicketFreeText, TicketFreeKey, TicketFreeTime, ArticleFreeText and ArticleFreeKey fields that were commonly used in OTRS 3.0 and before. The limitation of these "Free Fields" was that they can be defined up to 16 (text or dropdown) fields and 6 time fields for a ticket and 3 (text or dropdown) fields for each article only, not more.

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

This new framework that handles the dynamic fields is 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.

Die folgenden dynamischen Feldtypen sind in diesem Release inbegriffen:

- Text (eine Zeile)
- Textfeld (mehrere Zeilen Text)
- Kontrollkästchen
- Einfachauswahl (eine Auswahl, mehrere Auswahlmöglichkeiten)
- Mehrfachauswahl (mehrere Auswahlmöglichkeiten)
- Datum
- Datum / Zeit

10.2. Konfiguration

Standardmäßig enthält eine Neu-Installation von OTRS keine dynamische Felder. Wenn Sie solche Felder in Tickets oder Artikel verwenden möchten, müssen Sie dynamische Felder erstellen.

Die Konfiguration dynamischer Felder teilt sich in zwei Bereiche auf. Um ein dynamisches Feld hinzuzufügen oder ein bereits existierendes zu editieren navigieren Sie im Admin-Interface auf den Link "Dynamische Felder". Um ein dynamisches Feld als verpflichtend zu setzen oder um es in den Oberflächen zu verstecken, müssen Sie die benötigten Einstellungen in der "Sysconfig" vornehmen.

10.2.1. Hinzufügen eines dynamischen Feldes

Click on the "Admin" button located in the navigation bar, then click on the "Dynamic Fields" link inside "Ticket Settings" box located in the lower center of the screen. The dynamic fields overview will display as follows:

Abbildung 4.81. Leere Übersichtsseite der dynamischen Felder

Dynamic Fields Management - Overview

Actions

Ticket

Add new field for object: Ticket

Article

Add new field for object: Article

Hint

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

Dynamic Fields List

NAME	LABEL	ORDER	TYPE	OBJECT	VALIDITY	DELETE
ProcessManagementProcessID	Process	1	ProcessID	Ticket	valid	
ProcessManagementActivityID	Activity	2	ActivityID	Ticket	valid	

1-2 of 2

Beachten Sie das sich diese Übersicht ändern wird, wenn Sie dynamische Felder hinzufügen. Diese Übersicht könnte bereits einige dynamische Felder anzeigen, wenn die Installation von einer älteren OTRS Version geupdatet wurde.

Die Aktionen in der linken Seitenleiste zeigt zwei Möglichkeiten an: Artikel und Ticket, jede Möglichkeit hat seine eigene Dropdown-Auswahl für dynamische Felder.

Anmerkung

Die Installation eines OTRS-Paketes kann mehrere Objekte zur SideBar hinzufügen.

Der allgemeine Ablauf um dynamische Felder zu erstellen ist:

- Click on the desired dynamic field object dropdown in the Actions side bar.
- Klicken Sie auf den Typ des dynamischen Feldes, den Sie aus der Liste hinzufügen wollen.
- füllen der Konfiguration.
- Speichern.

Der Konfigurationsdialoge für die dynamischen Felder sind in zwei Bereiche unterteilt. Der obere Teil ist für alle Felder gleich und der untere Teil ist für jeden Feldtyp unterschiedlich.

Allgemeine Einstellungen für dynamische Felder:

- Name: verpflichtend, einmalig, es sind nur Buchstaben und Zahlen erlaubt.

Das ist der interne Name des Feldes um zum Beispiel ein Feld in einer Oberfläche anzeigen oder verstecken zu lassen. Jede Veränderung eines Feldnamens (nicht empfohlen) benötigt ebenfalls einen manuellen Eingriff in den "Sysconfig"-Einstellungen, in denen das Feld referenziert wird.

- Bezeichnung: verpflichtend.

Dieser Name wird auf den Bildschirmen angezeigt, auf denen dieses Feld aktiv ist, es werden Übersetzungen unterstützt.

Anmerkung

Übersetzungen der Beschriftungen müssen Sie manuell zu den Übersetzungsdateien hinzufügen.

- Feldreihenfolge: verpflichtend.

Definiert die relative Reihenfolge, in der das Feld in Oberflächen angezeigt wird. Standardmäßig werden neue Felder an der letzten Position angezeigt. Wird diese Einstellung geändert, betrifft das ebenfalls die Reihenfolge anderer dynamischen Felder.

- Gültigkeit: verpflichtend.

Ein ungültiges dynamisches Feld wird in keiner Oberfläche angezeigt, gan egal ob es als sichtbar konfiguriert ist.

- Feldtyp: verpflichtet, nur lesend.

Zeigt den aktuell ausgewählten Feldtyp an.

- Objekttyp: verpflichtend, nur lesend.

Zeigt den Geltungsbereich des Feldes.

Anmerkung

Um die spezifischen Feldeinstellungen zu verdeutlichen, werden einige dynamische Felder in unserem Beispiel hinzugefügt. Diese neuen dynamischen Felder werden in späteren Abschnitten referenziert.

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 choose the field from the Article dropdown list.

Tabelle 4.6. Die folgenden Felder werden zu dem System hinzugefügt:

Name	Beschriftung	Typ
Feld1	Mein Feld 1	Text
Feld2	Mein Feld 2	Textfeld
Feld3	Mein Feld 3	Kontrollkästchen
Feld4	Mein Feld 4	Einfachauswahl
Feld5	Mein Feld 5	Mehrfachauswahl
Feld6	Mein Feld 6	Datum
Feld7	My Field 7	Datum / Zeit

10.2.2. Konfiguration der Textzeilen in dynamischen Feldern

dynamische Felder vom Typ Text wird benutzt und Einfachzeilen zu speichern.

Einstellungen für Texte in dynamischen Feldern:

- Standardwert: Optional

Dieser Wert wird standardmäßig in den Editieroberflächen angezeigt (wie Neues Telefenticket oder Verfassen).

- Link anzeigen: Optional.

Wenn gesetzt, wird der Feldwert in einen klickbaren Link in den Oberflächen konvertiert (wie TicketZoom oder Übersichten).

Zum Beispiel, wenn "Show link" auf "http://www.otrs.com" gesetzt ist, wird ein klick auf diesen Wert die OTRS-Webseite in Ihrem Browser aufrufen.

Anmerkung

The use of [% Data.NameX | uri %] in the Set link value, where NameX is the name of the field, will add the field value as part of the link reference.

Abbildung 4.82. Konfigurationsdialog für Textfelder in dynamischen Feldern

Dynamic Fields - Ticket: Add Text Field

Actions

Go back to overview

General

★ **Name:** Validity:
Must be unique and only accept alphabetic and numeric characters.

★ **Label:** Field type:
This is the name to be shown on the screens where the field is active.

★ **Field order:** Object type:
This is the order in which this field will be shown on the screens where is active.

Text Field Settings

Default value:
This is the default value for this field.

Show link:
Here you can specify an optional HTTP link for the field value in Overviews and Zoom screens.
Example: http://some.example.com/handle?query=[% Data.Field1 | uri %]

Check RegEx:
Here you can specify a regular expression to check the value. The regex will be executed with the modifiers xms.
Example: ^[0-9]\$

Add RegEx:

or

10.2.3. Konfiguration von Textfeldern in dynamischen Feldern

dynamische Felder vom Typ Textbereich werden verwendet um Mehrfachzeilen zu speichern.

Einstellungen für Textfelder in dynamischen Feldern:

- Anzahl der Zeilen: Optional, Zahl.

Verwendet, um die Höhe des Feldes in den Editier-Oberflächen (wie Neues Telefonticket oder TicketCompose) zu definieren.

- Anzahl der Spalten: Optional, Integer.

Dieser Wert wird verwendet, um die Breite des Feldes in den Editier-Oberflächen zu definieren.

- Standardwert: Optional

Dies ist der Wert, der standardmäßig in den Editier-Oberflächen angezeigt wird (kann ein mehrzeiliges Textfeld sein).

Abbildung 4.83. Konfigurationsdialog für Textfelder in dynamischen Feldern

Dynamic Fields - Ticket: Change Textarea Field

Actions

Go back to overview

General

★ Name: Validity:
Must be unique and only accept alphabetic and numeric characters.

★ Label: Field type:
This is the name to be shown on the screens where the field is active.

★ Field order: Object type:
This is the order in which this field will be shown on the screens where is active.

Textarea Field Settings

Number of rows:
Specify the height (in lines) for this field in the edit mode.

Number of cols:
Specify the width (in characters) for this field in the edit mode.

Default value:

This is the default value for this field.

Check RegEx: Here you can specify a regular expression to check the value. The regex will be executed with the modifiers xms.
Example: ^[0-9]\$\

Add RegEx:

Submit or Cancel

10.2.4. Konfiguration von Checkboxen in dynamischen Feldern

Checkbox in dynamischen Feldern werden verwendet, um wahre oder falsche Werte, dargestellt durch ein aktiviertes oder deaktiviertes Kontrollkästchen, zu speichern.

Einstellungen für Checkboxen in dynamischen Feldern:

- Standardwert: Verpflichtend.

Dies ist der Wert, der standardmäßig in den Editier-Oberflächen angezeigt wird (kann ein mehrzeiliges Textfeld sein), der Standard-Wert für dieses Feld kann 'Aktiviert' oder 'Deaktiviert' sein.

Abbildung 4.84. Konfigurationsdialog für Checkboxen in dynamischen Feldern

Dynamic Fields - Ticket: Add Checkbox Field

Actions

Go back to overview

General

★ Name: Validity:
Must be unique and only accept alphabetic and numeric characters.

★ Label: Field type:
This is the name to be shown on the screens where the field is active.

★ Field order: Object type:
This is the order in which this field will be shown on the screens where is active.

Checkbox Field Settings

Default value:
This is the default value for this field.

Submit or Cancel

10.2.5. Konfiguration von Einfachauswahlen in dynamischen Feldern

Dynamische Felder vom Typ Einfachauswahl werden verwendet um einen einfachen Wert, aus einer Liste, zu speichern.

Einstellungen für Einfachauswahl in dynamischen Feldern:

- Mögliche Werte: Verbindlich.

Liste aus Werten, die ausgewählt werden können. Soll ein zusätzliches Listen-Item hinzugefügt werden, ist es nötig den Key (interner Wert) und den Wert (angezeigter Wert) anzugeben.

- Standardwert: Optional

Dies ist der Wert, der standardmäßig in den Editier-Oberflächen angezeigt wird (wie Neues Telefonticket oder Verfassen), der Standard-Wert für dieses Feld kann aus den definierten Werten vorgegeben werden.

- Hinzufügen eines leeren Wertes: Verpflichtend, (Ja / Nein).

Wenn diese Option aktiviert ist, wird ein extra Wert hinzugefügt, der als "-" in der Liste der möglichen Werte angezeigt wird. Dieser spezielle Wert ist intern als leer gekennzeichnet.

- Wertübersetzung: Verpflichtend, (Ja / Nein).

Diese Einstellung wird verwendet um die möglichen Werte dieses Feldes zu übersetzen. Es werden jedoch nur die angezeigten Werte übersetzt, die internen Werte sind nicht davon betroffen. Die Übersetzung der Werte müssen manuell in den Übersetzungsdateien hinzugefügt werden.

- Link anzeigen: Optional.

Wenn gesetzt, wird der Feldwert in einen klickbaren Link in den Oberflächen konvertiert (wie TicketZoom oder Übersichten).

Zum Beispiel, wenn "Show link" auf "http://www.otrs.com" gesetzt ist, wird ein klick auf diesen Wert die OTRS-Webseite in Ihrem Browser aufrufen.

Anmerkung

The use of [% Data.NameX | uri %] in the Set link value, where NameX is the name of the field, will add the field value as part of the link reference.

Abbildung 4.85. Konfigurationsdialog für Einfachauswahl in dynamischen Feldern

Dynamic Fields - Ticket: Add Dropdown Field

Actions

Go back to overview

General

★ **Name:** Validity:

Must be unique and only accept alphabetic and numeric characters.

★ **Label:** Field type:

This is the name to be shown on the screens where the field is active.

★ **Field order:** Object type:

This is the order in which this field will be shown on the screens where is active.

Dropdown Field Settings

Possible values: ★ **Key:** ★ **Value:**

Activate this option to create an empty selectable value.

Add empty value:

Activate this option to display values as a tree.

Tree View:

If you activate this option the values will be translated to the user defined language. Note: You need to add the translations manually into the language translation files.

Translatable values:

Show link:

Here you can specify an optional HTTP link for the field value in Overviews and Zoom screens. Example: <http://some.example.com/handle?query={% Data.Field1 | uri %}>

Submit or Cancel

10.2.6. Konfiguration der Mehrfachauswahl in dynamischen Feldern

Dynamische Felder vom Typ Mehrfachauswahl werden verwendet um mehrere Werte aus einer definierten Liste zu speichern.

Einstellungen für dynamische Felder vom Typ Mehrfachauswahl:

- Mögliche Werte: Verbindlich.

Liste aus Werten, die ausgewählt werden können. Soll ein zusätzliches Listen-Item hinzugefügt werden, ist es nötig den Key (interner Wert) und den Wert (angezeigter Wert) anzugeben.

- Standardwert: Optional

Dies ist der Wert, der standardmäßig in den Editier-Oberflächen angezeigt wird (wie Neues Telefonticket oder Verfassen), der Standard-Wert für dieses Feld kann aus den definierten Werten vorgegeben werden.

- Hinzufügen eines leeren Wertes: Verpflichtend, (Ja / Nein).

Wenn diese Option aktiviert ist, wird ein extra Wert hinzugefügt, der als "-" in der Liste der möglichen Werte angezeigt wird. Dieser spezielle Wert ist intern als leer gekennzeichnet.

- Wertübersetzung: Verpflichtend, (Ja / Nein).

Diese Einstellung wird verwendet um die möglichen Werte dieses Feldes zu übersetzen. Es werden jedoch nur die angezeigten Werte übersetzt, die internen Werte sind nicht

davon betroffen. Die Übersetzung der Werte müssen manuell in den Übersetzungsdateien hinzugefügt werden.

Abbildung 4.86. Konfigurationsdialog für Mehrfachauswahlen in dynamischen Feldern

Dynamic Fields - Ticket: Add Multiselect Field

Actions

Go back to overview

General

★ Name: Validity:
Must be unique and only accept alphabetic and numeric characters.

★ Label: Field type:
This is the name to be shown on the screens where the field is active.

★ Field order: Object type:
This is the order in which this field will be shown on the screens where is active.

Multiselect Field Settings

Possible values:

★ Key: <input type="text" value="A"/>	★ Value: <input type="text" value="Option A"/>	<input type="checkbox"/>
★ Key: <input type="text" value="B"/>	★ Value: <input type="text" value="Option B"/>	<input type="checkbox"/>
★ Key: <input type="text" value="C"/>	★ Value: <input type="text" value="Option C"/>	<input type="checkbox"/>

Add value:

Default value:
This is the default value for this field.

Add empty value:
Activate this option to create an empty selectable value.

Tree View:
Activate this option to display values as a tree.

Translatable values:
If you activate this option the values will be translated to the user defined language.
Note: You need to add the translations manually into the language translation files.

or

10.2.7. Konfiguration von Datumsfeldern in dynamischen Feldern

Dynamische Felder vom Typ Datum werden verwendet um ein Datum zu speichern. (Tag, Monat und Jahr).

Einstellungen für Datumsfelder in dynamischen Feldern:

- Standard-Datumsunterschied: Optional, Integer.

Sekunden (positiv oder negativ) zwischen dem aktuellen Datum und dem gewählten Datum, dass standardmäßig in Editieroberflächen gewählt sein soll (wie Neues Telefongticket oder Verfassen).

- Define years period: Mandatory, (Yes / No).

Used to set a defined number of years in the past and the future based on the current date of the year select for this field. If set to Yes the following options are available:

- Jahre in der Vergangenheit: Optional, positive Zahlen.

Definiert die Anzahl der Jahre in der Vergangenheit die angezeigt werden sollen, ausgehend vom aktuellen Datum.

- Jahre in der Zukunft: Optional, positive Zahlen

Definiert die Anzahl der Jahre in der Zukunft die angezeigt werden sollen, ausgehend vom aktuellen Datum.

- Link anzeigen: Optional.

Wenn gesetzt, wird der Feldwert in einen klickbaren Link in den Oberflächen konvertiert (wie TicketZoom oder Übersichten).

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 [% Data.NameX | uri %] in the Set link value, where NameX is the name of the field will add the field value as part of the link reference.

Abbildung 4.87. Konfigurationsdialog für Datum in dynamischen Feldern.

Dynamic Fields - Ticket: Add Date Field

Actions

◀ Go back to overview

General

★ **Name:** Validity:
Must be unique and only accept alphabetic and numeric characters.

★ **Label:** Field type:
This is the name to be shown on the screens where the field is active.

★ **Field order:** Object type:
This is the order in which this field will be shown on the screens where is active.

Date Field Settings

Default date difference:
The difference from NOW (in seconds) to calculate the field default value (e.g. 3600 or -60).

Define years period:
Activate this feature to define a fixed range of years (in the future and in the past) to be displayed on the year part of the field.

Show link:
Here you can specify an optional HTTP link for the field value in Overviews and Zoom screens.
Example: http://some.example.com/handle?query=[% Data.Field1 | uri %]

Restrict entering of dates:
Here you can restrict the entering of dates of tickets.

or

10.2.8. Konfiguration von Datum / Zeit in dynamischen Feldern

Dynamische Felder vom Typ Datum / Zeit werden verwendet um ein Datum und eine Zeit zu speichern. (Minute, Stunde, Tag, Monat und Jahr).

Einstellung für Datum- / Zeitfelder in dynamischen Feldern:

- Standard-Datumsunterschied: Optional, Integer.

Sekunden (positiv oder negativ) zwischen dem aktuellen Datum und dem gewählten Datum, dass standardmäßig in Editieroberflächen gewählt sein soll (wie Neues Telefonticket oder Verfassen).

- Define years period: Mandatory, (Yes / No).

Used to set a defined number of years in the past and the future based on the current date of the year select for this field. If set to Yes the following options are available:

- Jahre in der Vergangenheit: Optional, positive Zahlen.

Definiert die Anzahl der Jahre in der Vergangenheit die angezeigt werden sollen, ausgehend vom aktuellen Datum.

- Jahre in der Zukunft: Optional, positive Zahlen

Definiert die Anzahl der Jahre in der Zukunft die angezeigt werden sollen, ausgehend vom aktuellen Datum.

- Link anzeigen: Optional.

Wenn gesetzt, wird der Feldwert in einen klickbaren Link in den Oberflächen konvertiert (wie TicketZoom oder Übersichten).

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 [% Data.NameX | uri %] in the Set link value, where NameX is the name of the field will add the field value as part of the link reference.

Abbildung 4.88. Konfigurationsdialog für Datum / Zeit in dynamischen Feldern

Dynamic Fields - Ticket: Add Date / Time Field

Actions

◀ Go back to overview

General

★ **Name:** Validity:

Must be unique and only accept alphabetic and numeric characters.

★ **Label:** Field type:

This is the name to be shown on the screens where the field is active.

★ **Field order:** Object type:

This is the order in which this field will be shown on the screens where is active.

Date / Time Field Settings

Default date difference: The difference from NOW (in seconds) to calculate the field default value (e.g. 3600 or -60).

Define years period: Activate this feature to define a fixed range of years (in the future and in the past) to be displayed on the year part of the field.

Show link: Here you can specify an optional HTTP link for the field value in Overviews and Zoom screens. Example: http://some.example.com/handle?query=[% Data.Field1 | uri %]

Restrict entering of dates: Here you can restrict the entering of dates of tickets.

or

10.2.9. Editieren eines dynamischen Feldes

Eine gefüllte Übersichtsseite der dynamischen Feldern (mit den vorherigen Beispielen) sollte so aussehen:

Abbildung 4.89. Übersichtsseite der dynamischen Felder mit Beispieldaten

Dynamic Fields Management - Overview

Actions

Ticket

Add new field for object: Ticket

Article

Add new field for object: Article

Hint

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

Dynamic Fields List 1-9 of 9

NAME	LABEL	ORDER	TYPE	OBJECT	VALIDITY	DELETE
ProcessManagementProcessID	Process	1	ProcessID	Ticket	valid	
ProcessManagementActivityID	Activity	2	ActivityID	Ticket	valid	
Field1	My field 1	3	Text	Ticket	valid	
Field2	My field 2	4	Textarea	Ticket	valid	
Field3	My field 3	5	Checkbox	Ticket	valid	
Field4	My field 4	6	Dropdown	Ticket	valid	
Field5	My field 5	7	Multiselect	Ticket	valid	
Field6	My field 6	8	Date	Ticket	valid	
Field7	My field 7	9	Date / Time	Ticket	valid	

Um ein dynamisches Feld zu ändern oder zu editieren, muss mindestens ein dynamisches Feld definiert sein. Wählen Sie ein bereits bestehendes Feld aus der Übersichtsseite aus und aktualisieren Sie die Einstellungen.

Anmerkung

Nicht alle Einstellungen von dynamischen Feldern können geändert werden. Der Feld- und Objekttyp können nicht verändert werden.

Es wird nicht empfohlen, den internen Feldnamen zu ändern, aber die Beschriftung kann jederzeit geändert werden. Wenn der interne Feldname geändert wird, müssen alle Einstellungen in der "SysConfig", die auf dieses Feld verweisen ebenfalls aktualisiert werden, genauso wie die Benutzereinstellungen (falls definiert).

10.2.10. Anzeigen eines dynamischen Feldern

Um ein dynamisches Feld auf einer Oberfläche anzuzeigen, sind zwei zwingende Voraussetzungen nötig:

1. Das dynamische Feld muss gültig sein.
2. Das dynamische Feld muss in der Konfiguration der Oberfläche auf 1 oder 2 gesetzt sein.

Folgen Sie diesen Schritten um ein dynamisches Feld in einer Oberfläche anzuzeigen

- Vergewissern Sie sich, dass das dynamische Wert auf gültig gesetzt ist. Sie sehen die Gültigkeit in der Übersichtsseite der dynamischen Felder. Setzen Sie, wenn nötig, das dynamische Feld auf gültig.
- Öffnen Sie die "Sysconfig" und wählen Sie in der Dropdownliste in der linken Aktionen-Sidebar "Ticket".

Anmerkung

Ebenfalls können Sie nach "DynamicField" oder nach dem Sysconfig-Schlüssel direkt suchen.

- Suchen Sie nach der Untergruppe für die Oberfläche, die Sie editieren wollen und klicken Sie darauf. Zum Beispiel: "Frontend::Agent::Ticket::ViewPhoneNew".
- Suchen Sie nach der Einstellung, die mit "###DynamicField" endet. Zum Beispiel: "Ticket::Frontend::AgentTicketPhone###DynamicField".

- Wenn die Einstellung leer ist oder nicht den benötigten Feldnamen besitzt, klicken Sie auf "+" um einen neuen Eintrag hinzuzufügen. Zum Beispiel: Field1, Content: 1.

Wenn diese Einstellung bereits das dynamische Feld anzeigt, vergewissern Sie sich das es auf "1" gesetzt ist, damit es angezeigt wird oder das es auf "2" gesetzt ist, damit es verpflichtend ist.

- Save the configuration by clicking on the "Update" button at the bottom of the screen and navigate to the screen where you want the field to be displayed.

10.2.10.1. Beispiele

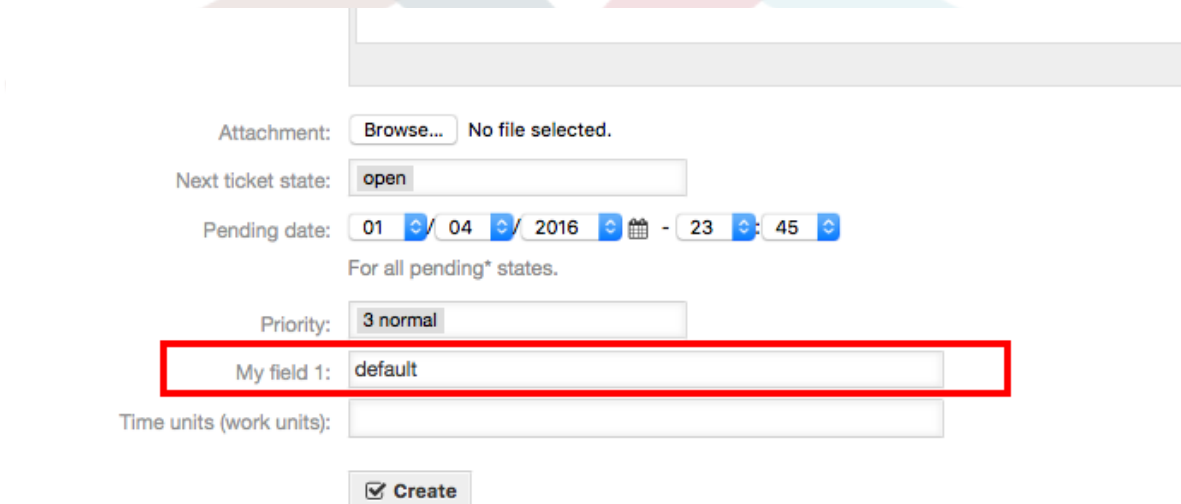
Die folgenden "SysConfig" Konfigurationen sind Beispiele um zu zeigen, wie dynamische Felder angezeigt oder versteckt werden können.

Beispiel 4.22. Feld1 im Dialog NewPhoneTicket anzeigen

- *Gruppe:* Ticket
- *Untergruppe:* Frontend::Agent::Ticket::ViewPhoneNew
- *Einstellung:* Ticket::Frontend::AgentTicketPhone###DynamicField
- *Wert:*

Schlüssel	Inhalt
Feld1	1

Abbildung 4.90. Field1 in New Phone Ticket Screen



Attachment: No file selected.

Next ticket state:

Pending date: / / :

For all pending* states.

Priority:

My field 1: default

Time units (work units):

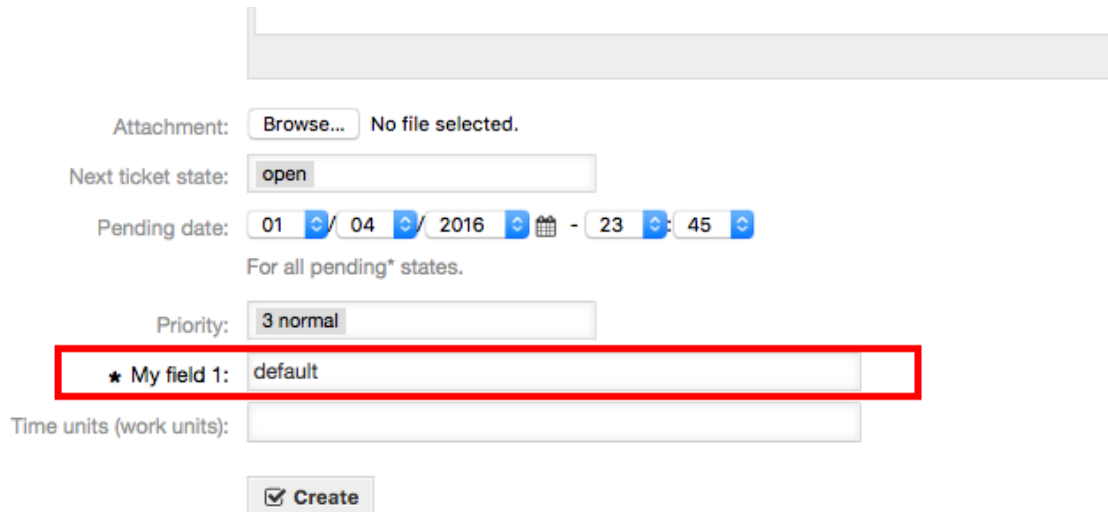
Beispiel 4.23. Activate Field1 in New Phone Ticket Screen as mandatory. Feld1 als verpflichtend in Dialog NewPhoneTicket setzen.

- *Gruppe:* Ticket
- *Untergruppe:* Frontend::Agent::Ticket::ViewPhoneNew
- *Einstellung:* Ticket::Frontend::AgentTicketPhone###DynamicField

- Wert:

Schlüssel	Inhalt
Feld1	2

Abbildung 4.91. Field1 in New Phone Ticket Screen as mandatory

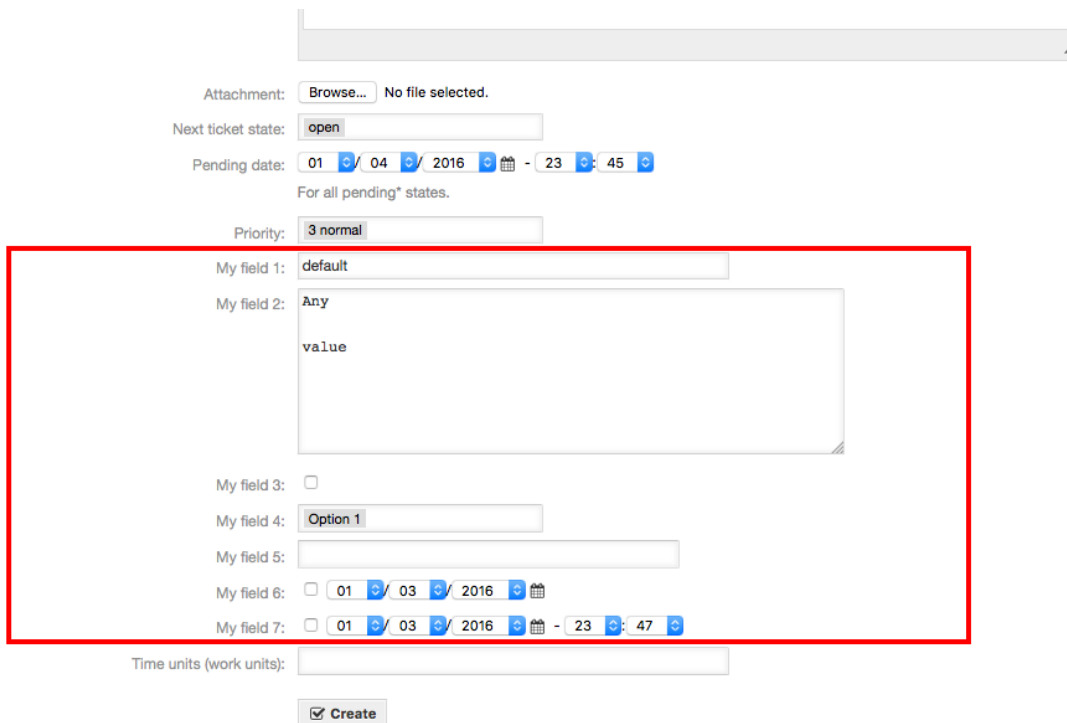


Beispiel 4.24. Aktivieren von verschiedenen Feldern in der Oberfläche 'Neues Telefonticket'.

- Gruppe: Ticket
- Untergruppe: Frontend::Agent::Ticket::ViewPhoneNew
- Einstellung: Ticket::Frontend::AgentTicketPhone###DynamicField
- Wert:

Schlüssel	Inhalt
Feld1	1
Feld2	1
Feld3	1
Feld4	1
Feld5	1
Feld6	1
Feld7	1

Abbildung 4.92. Several fields in New Phone Ticket Screen as mandatory



Attachment: No file selected.

Next ticket state:

Pending date: 01 / 04 / 2016 - 23 : 45

For all pending* states.

Priority:

My field 1:

My field 2:
value

My field 3:

My field 4:

My field 5:

My field 6: 01 / 03 / 2016

My field 7: 01 / 03 / 2016 - 23 : 47

Time units (work units):

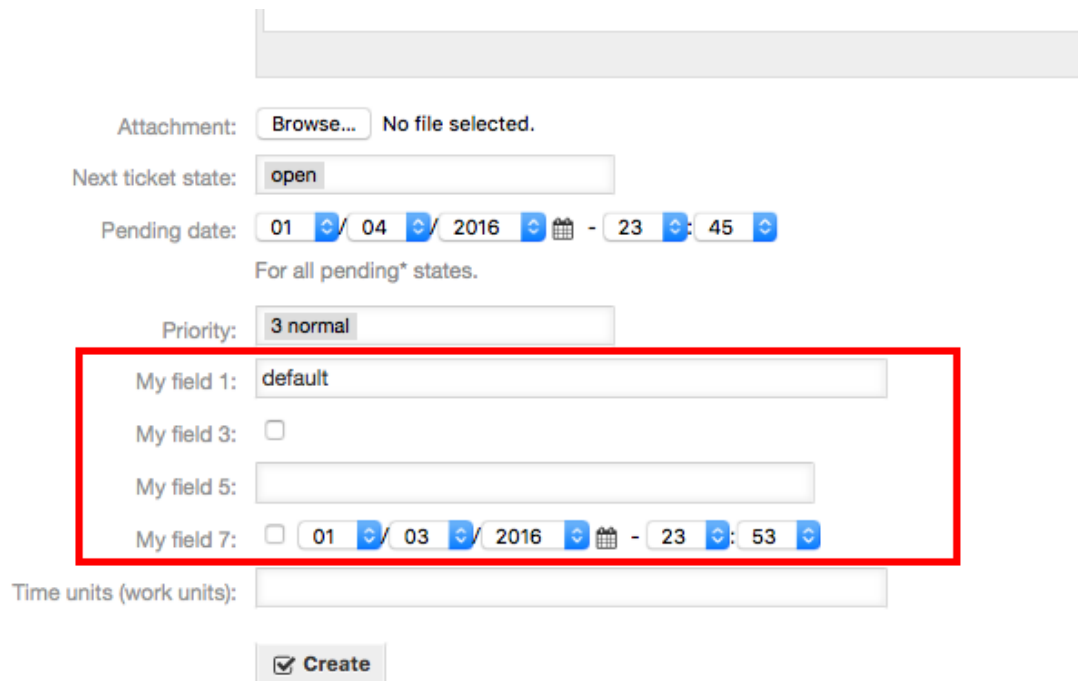
Create

Beispiel 4.25. Deaktivieren von Feldern in der Oberfläche 'Neues Telefonticket'.

- *Gruppe:* Ticket
- *Untergruppe:* Frontend::Agent::Ticket::ViewPhoneNew
- *Einstellung:* Ticket::Frontend::AgentTicketPhone###DynamicField
- *Wert:*

Schlüssel	Inhalt
Feld1	1
Feld2	0
Feld3	1
Feld4	0
Feld5	1
Feld6	0
Feld7	1

Abbildung 4.93. Some deactivated fields in New Phone Ticket Screen as mandatory

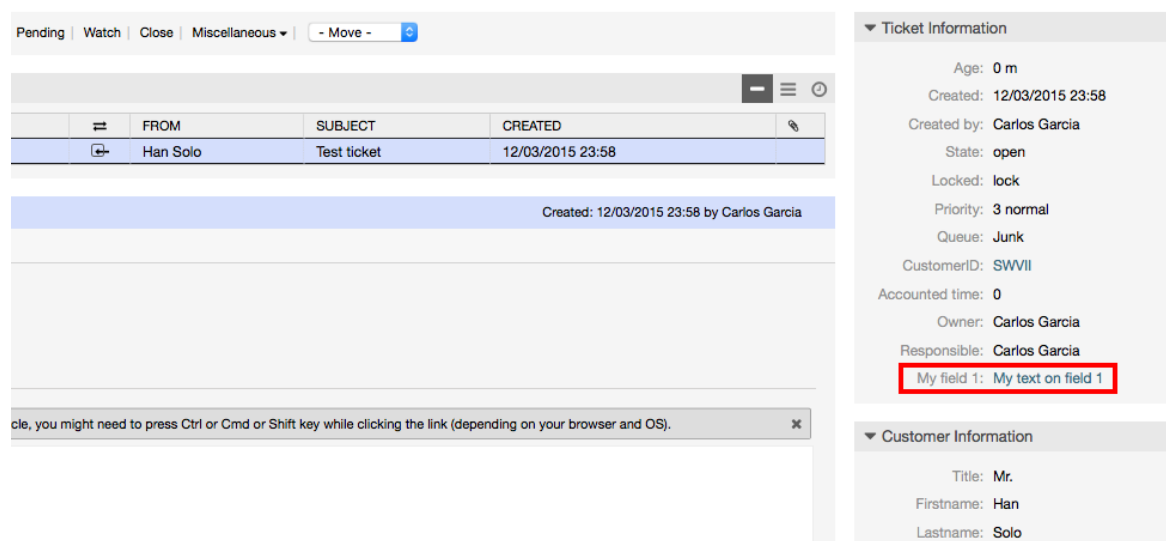


Beispiel 4.26. Aktiviert Feld1 in der TicketZoom-Oberfläche.

- *Gruppe:* Ticket
- *Untergruppe:* Frontend::Agent::Ticket::ViewZoom
- *Einstellung:* Ticket::Frontend::AgentTicketZoom###DynamicField
- *Wert:*

Schlüssel	Inhalt
Feld1	1

Abbildung 4.94. Field1 in Ticket Zoom Screen



FROM	SUBJECT	CREATED
Han Solo	Test ticket	12/03/2015 23:58

Ticket Information

Age: 0 m
 Created: 12/03/2015 23:58
 Created by: Carlos Garcia
 State: open
 Locked: lock
 Priority: 3 normal
 Queue: Junk
 CustomerID: SWII
 Accounted time: 0
 Owner: Carlos Garcia
 Responsible: Carlos Garcia
My field 1: My text on field 1

Customer Information

Title: Mr.
 Firstname: Han
 Lastname: Solo

Beispiel 4.27. Aktiviert Feld1 in der kleinen Ticketübersicht.

- *Gruppe:* Ticket
- *Untergruppe:* Frontend::Agent::TicketOverview
- *Einstellung:* Ticket::Frontend::OverviewSmall###DynamicField
- *Wert:*

Schlüssel	Inhalt
Feld1	1

Abbildung 4.95. Field1 in Ticket Overview Small Screen

QueueView: Junk

My Queues (2) Junk (1) Raw (1)

All tickets 2 Available tickets 1

Bulk 1-1 of 1 S M L

TICKET#	▲ AGE	FROM / SUBJECT	STATE	LOCK	QUEUE	MY FIELD 1	OWNER	CUSTOMERID
2016010354000013	11 m	Han Solo Test ticket	open	lock	Junk	My text on field 1	Carlos Garcia	SWVII

Diese Einstellung betrifft: Eskalationsübersichten, Übersicht über gesperrte Tickets, Queue-Übersichten, Übersicht über Verantwortlichkeiten, Status-Übersichten und Beobachten-Übersichten.

10.2.11. Setzen eines Standard-Wertes durch ein TicketEvent-Module

Ein Ticket-Event (z.B.: TicketCreate) kann einen Wert für ein bestimmtes Feld triggern, wenn das Feld noch keinen Wert besitzt.

Anmerkung

Bei Verwendung dieser Methode ist der Standard-Wert nicht sichtbar in der Editieroberfläche (z.B.: Neues Telefonticket), weil der Wert nach der Erstellung des Tickets gesetzt wird.

Um diese Funktion zu aktivieren ist es notwendig die folgende Einstellung zu aktivieren: "Ticket::EventModulePost###TicketDynamicFieldDefault".

Beispiel 4.28. Aktiviert Feld1 im Event TicketCreate.

- *Gruppe:* Ticket
- *Untergruppe:* Core::TicketDynamicFieldDefault
- *Einstellung:* Ticket::TicketDynamicFieldDefault###Element1

Anmerkung

Diese Konfiguration kann in jeder der 16 Ticket::TicketDynamicFieldDefault###Element Einstellungen getätigt werden.

Werden mehr als 16 Felder benötigt, müssen Sie sich eine eigene XML erstellen und diese im Verzeichnis \$OTRS_HOME/Kernel/Config/files ablegen, um dieses Feature zu erweitern.

- Wert:

Schlüssel	Inhalt
Ereignis	TicketCreate
Name	Feld1
Wert	Ein neuer Wert

10.2.12. Setzt die Standard-Einstellung abhängig von den Benutzereinstellungen

Der Standardwert der dynamischen Felder kann mit einem benutzerspezifischen Wert, der in den Benutzereinstellungen definiert ist, überschrieben werden

Bei Verwendung, wird der Standardwert des Feldes in jeder Oberfläche angezeigt, in der das Feld aktiviert ist (solange das Feld nicht bereits einen unterschiedlichen Wert besitzt).

Die "Sysconfig"-Einstellung "PreferencesGroups###DynamicField" ist in der Untergruppe "Frontend::Agent::Preferences" zu finden. Diese Einstellung ist ein Beispiel, wie man einen Eintrag in den Benutzereinstellungen erstellt, um einen exklusiven Standardwert des dynamischen Feldes für den ausgewählten Benutzer zu setzen. Die Einschränkung dieser Einstellung ist, dass nur die Verwendung eines dynamischen Feldes erlaubt ist. Wenn zwei oder mehr Felder dieses Feature verwenden, ist es notwendig eine benutzerdefinierte XML-Konfigurationsdatei zu erstellen, um weitere ähnliche Einstellungen wie diese hinzu zufügen.

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.

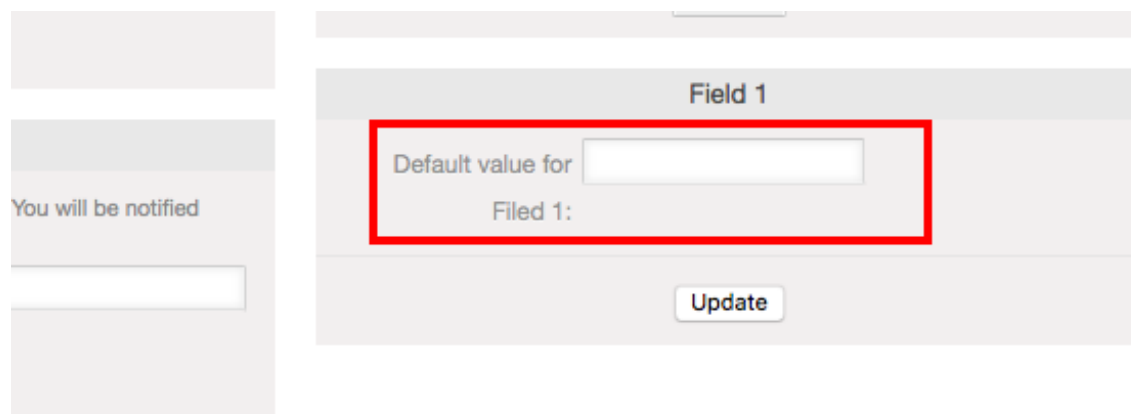
Beispiel 4.29. Aktiviert Field1 in den Benutzereinstellungen

- Gruppe: Ticket
- Untergruppe: Frontend::Agent::Preferences
- Einstellung: PreferencesGroups###101-DynamicField-Field1
- Wert:

Schlüssel	Inhalt
Ereignis	TicketCreate
Aktiv	1
Block	Eingabe
Spalte	Andere Einstellungen
Daten:	[% Env("UserDynamicField_Field1") %]
Schlüssel:	Mein Feld 1
Beschriftung:	Standardwert für: Mein Feld 1
Modul:	Kernel::Output::HTML::PreferencesGeneric

Schlüssel	Inhalt
PrefKey:	UserDynamicField_Field1
Priorität:	7000

Abbildung 4.96. Field1 in User preferences screen



11. Generic Interface

The OTRS Generic Interface consists of a multiple layer framework that lets OTRS communicate with other systems via a web service. This communication could be bi-directional:

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

11.1. Generische Schnittstellenebenen

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

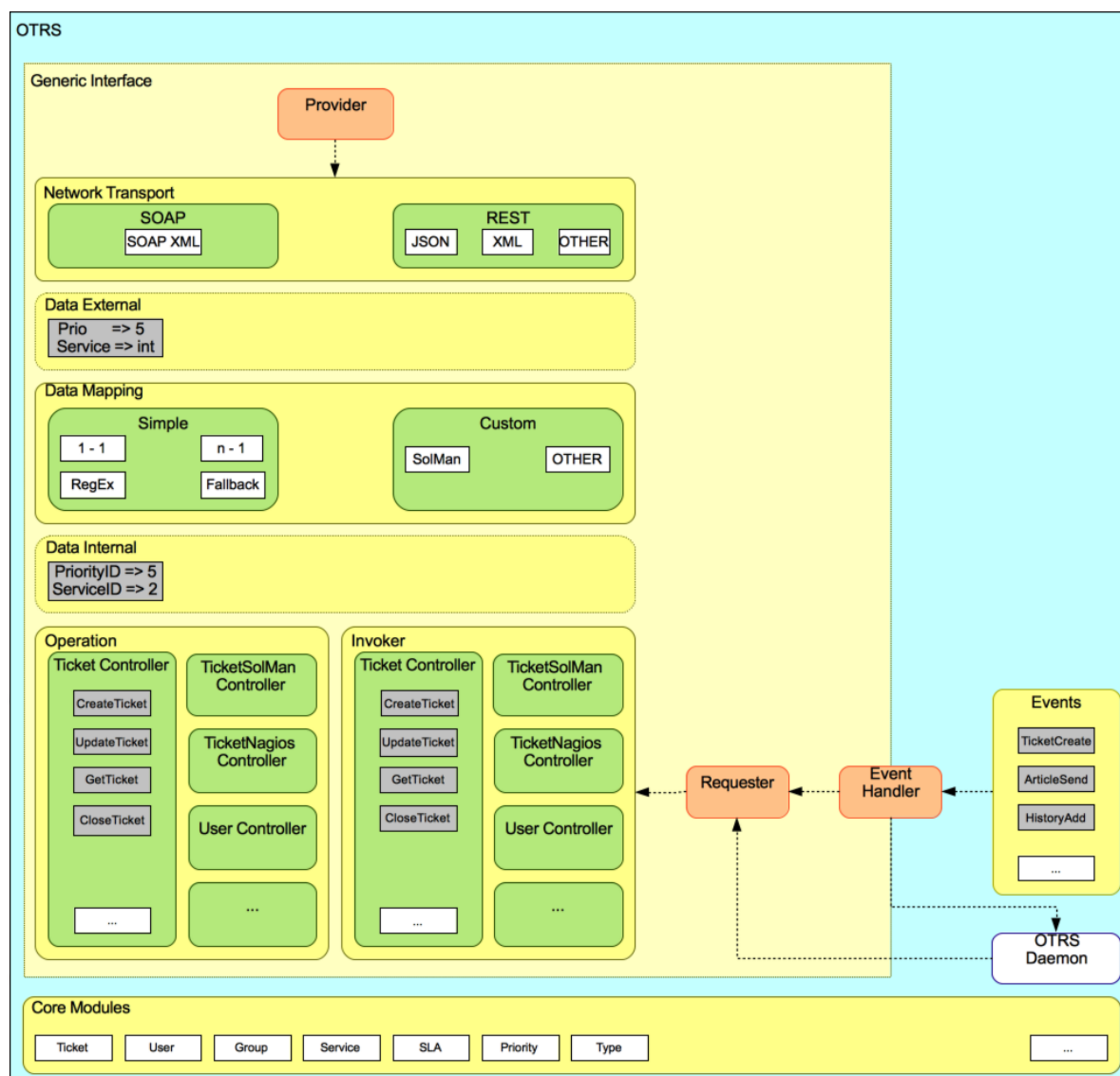
A layer is a set of files, which control how the Generic Interface performs different parts of a web service. Using the right configuration, one can build different web services for different External Systems without creating new modules.

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.

Abbildung 4.97. The graphical interface layers



11.1.1. Netzwerk-Transport

This layer is responsible for the correct communication with the Remote System. It receives requests and generates responses when acting as provider, and generates requests and receives responses when acting as requester.

Provider communication is handled by a new web server handle called "nph-genericinterface.pl".

Requester communication could be initiated during an event triggered by a Generic Interface module or any other OTRS module. This event is caught by the event handler and depending on the configuration the event will be processed directly by the requester object or delegated to the Scheduler (a separated daemon designed to process tasks asynchronously).

11.1.2. Daten-Mapping

This layer is responsible for translating data structures between OTRS and the Remote System (data internal and data external layers). Usually Remote Systems have different data structures than OTRS (including different values and names for those values), and

here resides the importance of the layer to change the received information into something that OTRS can understand and on the opposite way send the information to each Remote System using their data dictionaries.

Example: "Priority" (OTRS) might be called "Prio" in a remote system and it could be that value "1 Low" (OTRS) should be mapped to "Information" on the remote system.

11.1.3. 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.

11.1.4. Operation (OTRS als 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.

11.1.5. Invoker (OTRS als 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.

11.2. Generic Interface Kommunikationsfluss

Das Generic Interface hat einen definierten Fluss um Aktionen als Provider und Requester auszuführen.

Diese Flüsse sind wie folgt beschrieben:

11.2.1. OTRS als Provider

11.2.1.1. Entfernte Anfrage:

1. HTTP-Anfrage

- OTRS empfängt HTTP-Anfragen und leitet sie durch die verschiedenen Schichten weiter.
- Die Provider-Modul ist für die Ausführung und Kontrolle dieser Maßnahmen vorhanden.

2. Netzwerk-Transport

- Das Netzwerktransportmodul dekodiert den Daten-Payload und trennt den Operationsnamen aus dem Rest der Daten.
- Der Operationsname und die Betriebsdaten werden an den Provider zurückgegeben

3. externe Daten

- Daten die vom Remote-System gesendet wurden (Das ist kein Modulbasierter Layer).

4. Mapping

- Die Daten wird aus dem Fremdsystem-Format in das interne OTRS Format, wie in der Mapping-Konfiguration für diesen Vorgang (Mapping für eingehende Anforderungsdaten) spezifiziert, umgewandelt.
- Die bereits transformierten Daten werden an den Provider zurückgegeben.

5. interne Daten

- Sobald die Daten, transformiert und aufbereitet wurden, werden diese an die Operation weitergeleitet. (Das ist kein Modulbasierter Layer).

6. Operation

- Empfängt und prüft Daten
- Führt eine Benutzerzugangskontrolle durch.
- Führt die Aktion aus.

11.2.1.2. OTRS-Antwort:

1. Operation

- Liefert das Ergebnis an den Provider.

2. interne Daten

- Von der Operation zurückgegebene Daten.

3. Mapping

- The data is transformed back to the Remote system format as specified in the mapping configuration (Mapping for outgoing response data).
- Die bereits transformierten Daten werden an den Provider zurückgegeben.

4. externe Daten

- Data as transformed and prepared to be passed to Network Transport as response.

5. Netzwerk-Transport

- Empfängt die Daten im Format des Remote-Systems.
- Constructs a valid response for this network transport type.

6. HTTP-Antwort

- 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).

11.2.2. OTRS als Requester

11.2.2.1. OTRS Anfrage:

1. Event-Trigger-Steuerungsprogramm

- Basierend auf der Konfiguration des Webservices wird entschieden ob der Request synchron oder asynchron ist.
 - synchron
 - A direct call to the Requester is made in order to create a new request and to pass it through the layers.
 - asynchron
 - Create a new Generic Interface (Requester) task for the OTRS Daemon (by delegating the request execution to the Scheduler Daemon, the user experience could be highly improved, otherwise all the time needed to prepare the request and the remote execution will be added to the OTRS Events that trigger those requests).
 - In its next cycle the OTRS daemon process reads the new task and creates a call to the Requester that will create a new request and then passes it through the layers.

2. Invoker

- Empfängt Daten von dem Event.
- Überprüft empfangene Daten (wenn benötigt).
- 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. interne Daten

- 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. externe Daten

- Data as transformed and prepared for sending to the remote system.

6. Netzwerk-Transport

- Receives the remote operation name and the data already transformed to the Remote System format from the requester.
- Konstruiert eine gültige Anfrage für die Netzwerkübertragung.

- Sends the request to the remote system and waits for the response.

11.2.2.2. Remote Antwort:

1. Netzwerkübertragung

- Empfängt die Antwort und dekodiert den Daten-Payload.
- Sendet die Daten zurück zu dem Requester.

2. externe Daten

- Data as received from the Remote System.

3. Mapping

- Die Daten wird aus dem Fremdsystem-Format in das interne OTRS Format, wie in der Mapping-Konfiguration für diesen Vorgang (Mapping für eingehende Anforderungsdaten) spezifiziert, umgewandelt.
- The already transformed data is returned to the requester.

4. interne Daten

- Sobald die Daten, transformiert und aufbereitet wurden, werden diese an die Requester zurückgeschickt.

5. Invoker

- Empfängt die zurückgegebenen Daten.
- Verarbeitet die Daten als speziell von jedem Invoker benötigt werden (inklusive Fehlerbehandlung falls vorhanden).
- Gibt das Ergebnis und die Daten des Invokers an den Requester.

6. Event Handler or OTRS Daemon

- Receives the data from the Requester. In the case of the OTRS Daemon this data might contain information to create a task in the future.

11.3. 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 it is defined what actions the web service can perform internally (Operation), what actions the OTRS request can perform Remote System (Invokers), how data is converted from one system to the other (Mapping), and over which protocol the communication will take place (Transport).

The Generic Interface is the framework that makes it possible to create Web Services for OTRS in a predefined way, using already made building blocks that are independent from each other and interchangeable.

11.4. Web Service - Grafische Oberfläche

Die grafische Benutzeroberfläche (GUI) des Webservice ist ein Werkzeug, dass es erlaubt komplexe Webservice-Konfigurationen mit einer benutzerfreundlichen und komfortablen Oberfläche zu erstellen. Es erlaubt folgendes:

- Erstellen und Löschen von Webservices
- Importieren und Exportieren von Konfigurationen (im YAML Dateiformat) für existierende Webservices.
- Zeige, exportiere und setze alte Konfigurationen für existierende Webservices in der Web Service History - Anzeige zurück.
- Verfolge alle Kommunikations-Protokolle für alle Webservices in der Debugger-Anzeige.

11.4.1. Webdienstübersicht

Der "Web-Services"-Link im Hauptbildschirm der Admin-Oberfläche (in der "System Administration"-Box) führt zum Webservices-Überblick-Bildschirm, wo alle Webservice-Konfigurationen verwaltet werden können. Neue Webservices können hinzugefügt oder schon vorhandene Konfigurationen geändert werden.

Jede Webservice-Konfigurationsanzeige hat im oberen Teil des Bildschirms einen Navigationspfad im "Brotkrumen"-Stil. Dieser Navigationspfad ist nützlich um genau zu wissen, in welchem Teil der Webservice-Konfiguration man sich befindet und erlaubt dem Nutzer jederzeit zu irgendeinem Teil des Konfigurationsprozesses zurückzuspringen (jegliche Änderungen werden nicht gespeichert).

Anmerkung

Drücken Sie den Knopf "Webservice hinzufügen" um einen neuen Webservice zu erstellen, und geben Sie die erforderlichen Informationen an.

Abbildung 4.98. Web services overview

GenericInterface Web Service Management - Overview

You are here: > Web Services

Actions		Web Service List					
<input type="button" value="Add web service"/>		NAME	DESCRIPTION	REMOTE SYSTEM	PROVIDER TRANSPORT	REQUESTER TRANSPORT	VALIDITY
		Webservice one	First webservice	Any description	HTTP::SOAP	HTTP::SOAP	valid
		Webservice two	-	-	-	-	valid

11.4.2. Webservice hinzufügen

The only required field in this part is the web service "Name" that needs to be unique in the system and can not be left empty. Other fields are also necessary for the configuration like the "Debug Threshold" and "Validity" but these fields are already populated with the default value for each list.

The default value for "Debug Threshold" is "debug". When configured in this manner all communication logs are registered in the database. Each subsequent Debug Threshold value is more restrictive and discards communication logs of lower order than the one set in the system.

Schwellenwerte der Fehlersuche (von niedrig nach hoch)

- Fehlersuche
- Info
- Notiz
- Error

Es ist ebenfalls möglich das Netzwerkübertragungsprotokoll für "OTRS als Provider" und "OTRS als Requester" zu definieren.

Klicken Sie auf "Speichern" um den neuen Webservice in der Datenbank zu registrieren oder klicken Sie auf "Abbrechen" um die Änderungen zu verwerfen. Anschließend werden Sie auf die Übersichtsseite der Webservices weitergeleitet.

Wenn Sie bereits eine Webservice Konfigurationsdatei im YAML Format haben, können sie auf der linken Seite den "Webservice importieren"-Button klicken. Weitere Informationen über das importieren von Webservices finden Sie im nächsten Abschnitt "Webservice ändern".

Anmerkung

To change or add more details to a web service, click on the web service name in the web service overview screen.

Abbildung 4.99. Web services add

GenericInterface Web Service Management - Add

You are here: > Web Services > New Web service

Actions

← Go to overview

⬇ Import web service

Hint

After you save the configuration you will be redirected again to the edit screen.
If you want to return to overview please click the "Go to overview" button.

General

Name: Debug threshold:

Description: Validity:

Remote system:

OTRS as provider

In provider mode, OTRS offers web services which are used by remote systems.

Settings

Network transport:

Operations

Operations are individual system functions which remote systems can request.

NAME	DESCRIPTION	CONTROLLER	INBOUND MAPPING	OUTBOUND MAPPING
No data found.				

OTRS as requester

In requester mode, OTRS uses web services of remote systems.

Settings

Network transport:

Invokers

Invokers prepare data for a request to a remote web service, and process its response data.

NAME	DESCRIPTION	CONTROLLER	INBOUND MAPPING	OUTBOUND MAPPING
No data found.				

Save

11.4.3. Webservice ändern

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:

- Webservice kopieren
- Webservice exportieren
- Webservice importieren.
- Konfigurationschronik.
- Webservice löschen

- Debugger.

Anmerkung

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

11.4.3.1. Webservice kopieren

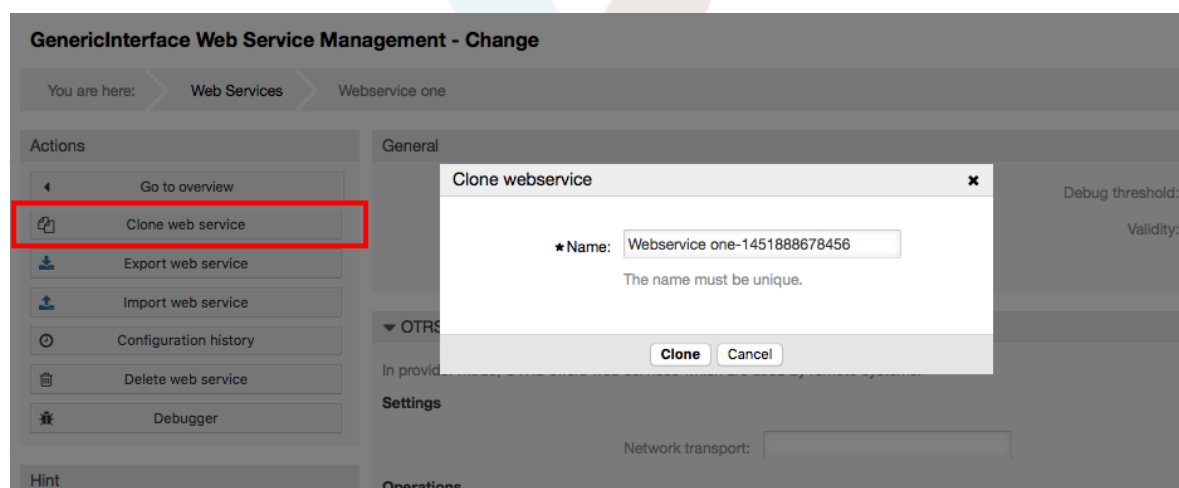
To clone a web service, you need to click on the "Clone web service" button. A dialog will be shown where you can use the default name or set a new name for the (cloned) web service.

Anmerkung

Remember that 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.

Abbildung 4.100. Webservice kopieren



11.4.3.2. Webservice exportieren

The "Export web service" button gives you the opportunity to dump the configuration of the current web service into a YAML file, to download it and to store it on your file system. This can be specially useful if you want to migrate the web service from one server to another, for example from a testing environment to a production system.

Warnung

Alle in der Webservice-Konfiguration gespeicherten Passwörter werden im Klartext exportiert.

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

Jeder Browser auf den verschiedenen Betriebssystemen hat seinen eigenen Speicher-Dialog und -Stil. Je nachdem welcher Browser mit welcher Konfiguration vorhanden ist, kann es sein, dass überhaupt kein Dialog angezeigt wird und die Datei direkt im Standard-Verzeichnis auf Ihrem Dateisystem gespeichert wird. Bitte überprüfen Sie Ihre Browser-Anleitung wenn Sie genauere Anweisungen benötigen.

Abbildung 4.101. Web services export

GenericInterface Web Service Management - Change

You are here: > Web Services > Webservice one

Actions

- Go to overview
- Clone web service
- Export web service
- Import web service
- Configuration history
- Delete web service
- Debugger

Hint

After you save the configuration you will be redirected again to the edit screen.
If you want to return to overview please click the "Go to overview" button.

General

Name: Webservice one

Opening Webservice one.yml

You have chosen to open:

- Webservice one.yml**
 which is: TXT file (213 bytes)
 from: **http://localhost**

What should Firefox do with this file?

- Open with
- Save File
- Do this automatically for files like this from now on.

11.4.3.3. Webservice importieren

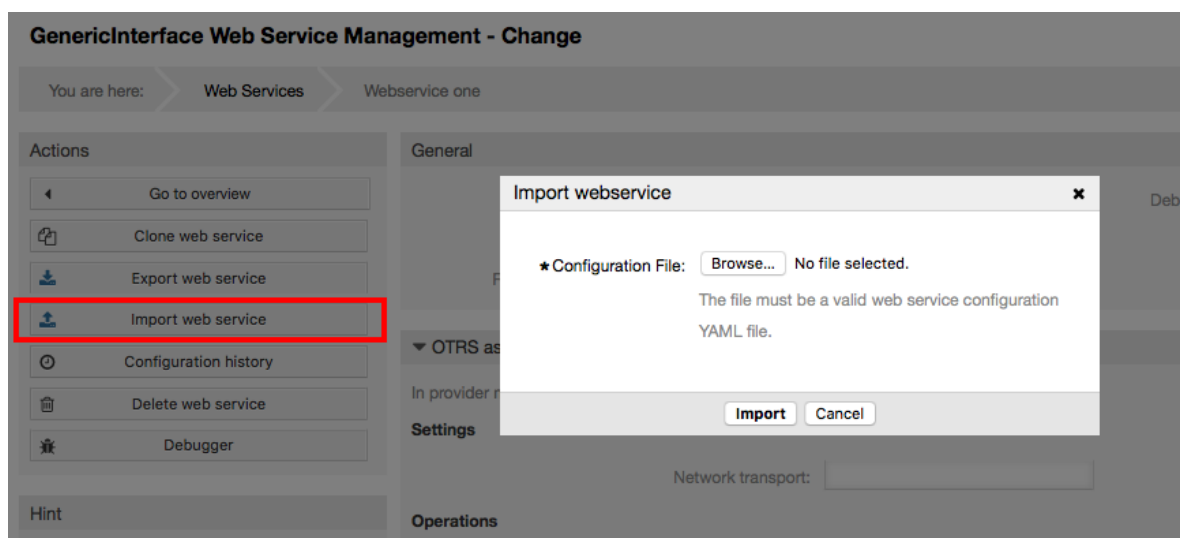
A valid web service configuration YAML file is required to use the import web service feature. Click on the "Import web service" button, browse for the configuration file or provide the complete path in the input box.

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.

Abbildung 4.102. Web services import



11.4.3.4. Webservice Historie

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 do not work as expected and it is not easy to revert the changes manually, you can click on the "Revert web service configuration" button. This will open a dialog to ask you if you are sure to revert the web service configuration. Click "Revert web service configuration" in this dialog to replace the current configuration with the selected version, or click "Cancel" to close the dialog.

Warnung

Remember that any passwords stored in the web service configuration will be exported in plain text format.

Please be careful when you restore a configuration because this process is irreversible.

Abbildung 4.103. Web service history

GenericInterface Configuration History for Web Service Webservice one

You are here: > Web Services > Webservice one > History

Actions

Go back to Web Service

Hint

Here you can view older versions of the current web service's configuration, export or even restore them.

Configuration History List

VERSION	CREATE TIME
11	2016-01-04 14:25:30
10	2016-01-04 14:25:26
9	2016-01-04 14:25:13
8	2016-01-04 14:25:10
7	2016-01-04 14:24:54

Select a single configuration version to see its details.

History Details: Version 11, 2016-01-04 14:25:30

Export web service configuration | Restore web service configuration

```

---
Debugger:
DebugThreshold: debug
TestMode: 0
Description: First webservice
Provider:
Operation:
CreateTicket:
  Description: ''
  MappingInbound: {}
  MappingOutbound: {}
  Type: Ticket::TicketCreate
GetTicket:
  Description: ''
  MappingInbound: {}
  MappingOutbound: {}
  Type: Ticket::TicketGet
SearchTicket:
  Description: ''
  MappingInbound: {}
  MappingOutbound: {}
  Type: Ticket::TicketSearch
UpdateTicket:
  Description: ''
  MappingInbound: {}
  MappingOutbound: {}
  Type: Ticket::TicketUpdate
Transport:
  Config:
    Authentication: {}
  Type: HTTP::SOAP
RemoteSystem: Any description
Requester:
  Transport:
    Config:
      Authentication: {}
    Type: HTTP::REST
          
```

11.4.3.5. Webservice löschen

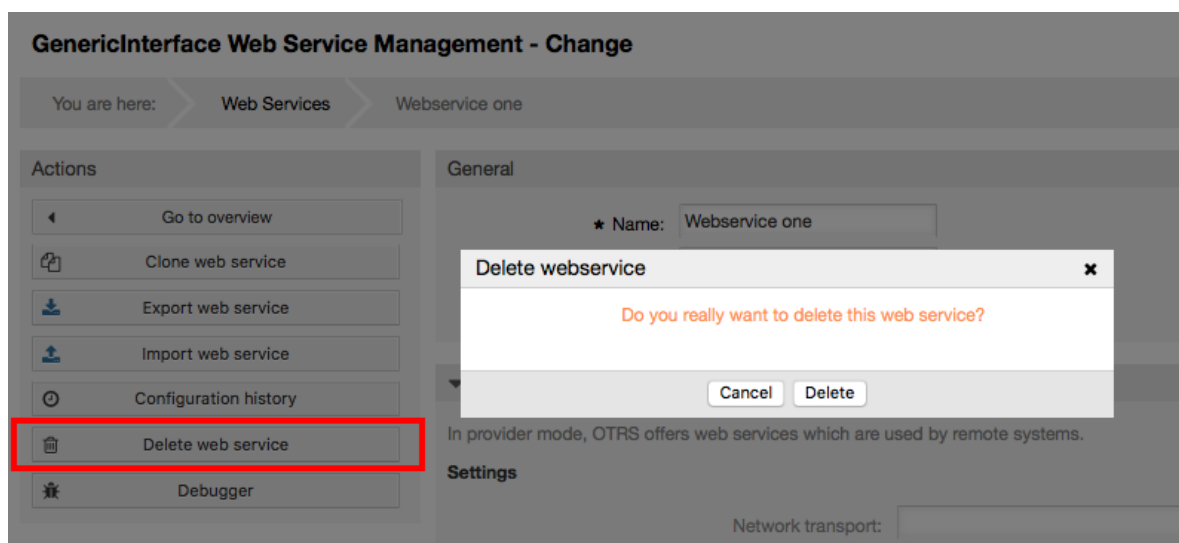
Sometimes it is necessary to delete a web service completely. To do this you can press on the "Delete web service" button and a new dialog will appear asking for confirmation.

Click on "Delete" to confirm the removal of the web service or on "Cancel" to close the dialog.

Warnung

Deleting a web service can't be undone, please be careful when deleting a web service.

Abbildung 4.104. Webservice löschen



11.4.3.6. Webservice 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:

- Kommunikationstyp (Provider oder Requester)
- Datum: vor und/oder nach einem bestimmten Datum
- Die Remote-IP-Adresse
- 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.

Abbildung 4.105. Web service debugger

GenericInterface Debugger for Web Service WebserviceOne

You are here: Web Services > WebserviceOne > Debugger

Actions

Go back to web service

Clear

Request List

Provider	2016-01-04 19:09:51	127.0.0.1
Provider	2016-01-04 19:10:57	127.0.0.1
Provider	2016-01-04 19:11:20	127.0.0.1
Provider	2016-01-04 19:11:20	127.0.0.1
Provider	2016-01-04 19:13:36	127.0.0.1
Provider	2016-01-04 19:14:14	127.0.0.1

Filter by type:

Filter from: 01 / 13 / 2015

Filter to: 01 / 04 / 2016

Filter by remote IP:

Limit: 100

Refresh

Select a single request to see its details.

Request Details

Communication sequence started (2016-01-04 19:14:14, debug)

Received data by provider from remote system (2016-01-04 19:14:14, debug)

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:tic="http://www.otrs.org/TicketConnector/">
  <soapenv:Header/>
  <soapenv:Body>
    <tic:TicketGet>
      <!--You have a CHOICE of the next 3 items at this level-->
      <!--Optional:-->
      <UserLogin></UserLogin>
      <!--Optional:-->
      <CustomerUserLogin?</CustomerUserLogin>
      <!--Optional:-->
      <SessionID>14j19y84EBwLGpseEVCs81U0N6kaeRpx</SessionID>
      <ChallengeToken>L0JUK0RgS76kDwYQTh5zTnrLdeYQG8yg</ChallengeToken>
      <!--Optional:-->
      <Password>test</Password>
      <!--1 or more repetitions:-->
      <TicketID>1</TicketID>
      <!--Optional:-->
      <DynamicFields?</DynamicFields>
      <!--Optional:-->
      <Extended?</Extended>
      <!--Optional:-->
      <AllArticles?</AllArticles>
      <!--Optional:-->
      <ArticleSenderType?</ArticleSenderType>
      <!--Optional:-->
      <ArticleOrder?</ArticleOrder>
      <!--Optional:-->
      <ArticleLimit?</ArticleLimit>
      <!--Optional:-->
      <Attachments?</Attachments>
    </tic:TicketGet>
  </soapenv:Body>
</soapenv:Envelope>
```

Detected operation 'TicketGet' (2016-01-04 19:14:14, debug)

No data provided

Incoming data before mapping (2016-01-04 19:14:14, debug)

```
$VAR1 = {
  'AllArticles' => '?',
  'ArticleLimit' => '?',
  'ArticleOrder' => '?',
  'ArticleSenderType' => '?',
  'Attachments' => '?',
  'ChallengeToken' => 'L0JUK0RgS76kDwYQTh5zTnrLdeYQG8yg',
  'CustomerUserLogin' => '?',
  'DynamicFields' => '?',
  'Extended' => '?',
  'OperationType' => 'TicketGet',
  'Password' => 'test',
  'SessionID' => '14j19y84EBwLGpseEVCs81U0N6kaeRpx',
  'TicketID' => '1',
  'UserLogin' => ''
};
```

Outgoing data before mapping (2016-01-04 19:14:15, debug)

Returning provider data to remote system (HTTP Code: 200) (2016-01-04 19:14:15, debug)

```
<?xml version="1.0" encoding="UTF-8"?><soap:Envelope soap:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
```

11.4.3.7. Ändern der Webservice Konfiguration

Returning to the web service change screen, now we are going to review the right side of it. Here we have the possibility to modify all the general data for a web service such as name, description, debug threshold, etc. Also there are two more sections below that allows us to modify specific parameters for communication types "OTRS as Provider" and "OTRS as Requester".

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

On the "OTRS as provider" section it is possible to set or configure the network transport protocol. Only network transport back-ends that are registered are shown on the list. To

configure the network transport click on the "Configure" button. It is also possible to add new operations in this box. To do this select one of the available operations from the "Add Operation" list. This will lead you to the operation configuration screen. After saving the new operation it will be listed in the table above.

"OTRS 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.

Abbildung 4.106. Web services change

GenericInterface Web Service Management - Change

You are here: **Web Services** > Webservice one

Actions

- Go to overview
- Clone web service
- Export web service
- Import web service
- Configuration history
- Delete web service
- Debugger

Hint

After you save the configuration you will be redirected again to the edit screen. If you want to return to overview please click the "Go to overview" button.

General

Name: Debug threshold:

Description: Validity:

Remote system:

OTRS as provider

In provider mode, OTRS offers web services which are used by remote systems.

Settings

Network transport:

Operations

Operations are individual system functions which remote systems can request.

NAME	DESCRIPTION	CONTROLLER	INBOUND MAPPING	OUTBOUND MAPPING
CreateTicket	-	Ticket::TicketCreate	-	-
GetTicket	-	Ticket::TicketGet	-	-
SearchTicket	-	Ticket::TicketSearch	-	-
UpdateTicket	-	Ticket::TicketUpdate	-	-

Add Operation:

OTRS as requester

In requester mode, OTRS uses web services of remote systems.

Settings

Network transport:

Invokers

Invokers prepare data for a request to a remote web service, and process its response data.

NAME	DESCRIPTION	CONTROLLER	INBOUND MAPPING	OUTBOUND MAPPING
No data found.				

Add Invoker:

Save

or or

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". If the configuration is re-visited, a new option "Save and Finish" will appear. The behavior of this feature is defined below.

"Save" will store the current configuration level in the database and it will return to the previous screen to review your changes or to configure deeper settings.

"Save and Finish" 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.

11.4.3.7.1. Webservice Requester Übertragung

In future the list of available network transports will be increased. Currently only "HTTP::SOAP" and "HTTP::REST" transports are available. Each transport has different configuration options to setup and they might use different frontend modules to configure them.

It is quite simple to configure the "HTTP::SOAP" protocol as provider. There are only two settings: "Namespace" and "Maximum message length". These fields are required. The first one is a URI to give SOAP methods a context, reducing ambiguities, and the second one is a field where you can specify the maximum size (in bytes) for SOAP messages that OTRS will process.

Abbildung 4.107. Web service provider network transport (HTTP::SOAP)

GenericInterface Transport HTTP::SOAP for Web Service Webservice one

You are here: > Web Services > Webservice one > Provider Transport HTTP::SOAP

Actions

Go back to web service

Network transport

Properties

Type: HTTP::SOAP

★ Namespace:
URI to give SOAP methods a context, reducing ambiguities.
 e.g um:otrs-com:soap:functions or http://www.otrs.com/GenericInterface/actions

★ Request name scheme:
Select how SOAP request function wrapper should be constructed.
 'FunctionName' is used as example for actual invoker/operation name.
 'FreeText' is used as example for actual configured value.

★ Response name scheme:
Select how SOAP response function wrapper should be constructed.
 'FunctionName' is used as example for actual invoker/operation name.
 'FreeText' is used as example for actual configured value.

★ Maximum message length:
Here you can specify the maximum size (in bytes) of SOAP messages that OTRS will process.

Sort options:

Add new first level element: Add

Outbound sort order for xml fields (structure starting below function name wrapper) - see documentation for SOAP transport.

Save or Save and finish or Cancel

For "HTTP::REST" the configuration might be a bit more complicated, as it grows dynamically for each configured operation by adding: "Route mapping for Operation '<OperationName>':" and "Valid request methods for Operation '<OperationName>':" settings to the default transport settings "Maximum message length:" and "Send Keep-Alive:"

- Route mapping for Operation '<OperationName>':

In this setting a resource path is set. This path must be defined according to the needs of the web service considering that the path in conjunction with the HTTP request method determines the Generic Interface operation to be executed.

Path can contain variables in the form of ':<VariableName>' each path string that fits on the position of the variable name will be added to the request payload using the variable name defined in this setting.

Beispiele:

Route mapping: /Resource

- Gültige Anfragen:

http://localhost/otrs/nph-genericinterface.pl/Webservice/Test/Resource

http://localhost/otrs/nph-genericinterface.pl/Webservice/Test/Resource?Param1=One

- Ungültige Anfragen:

`http://localhost/otrs/nph-genericinterface.pl/Webservice/Test/Resource/`

`http://localhost/otrs/nph-genericinterface.pl/Webservice/Test/Resource/OtherResource`

`http://localhost/otrs/nph-genericinterface.pl/Webservice/Test/Resource/OtherResource?Param1=One`

Route mapping: `/Resource/:ID`

- Gültige Anfragen:

`http://localhost/otrs/nph-genericinterface.pl/Webservice/Test/Resource/1`

`http://localhost/otrs/nph-genericinterface.pl/Webservice/Test/Resource/1?Param1=One`

In both cases ID = 1 will be sent to the operation as part of the payload. In the second case also Param1 = One will be added, depending on the HTTP request method other parameters will be added if they come as a JSON string in the request header.

- Ungültige Anfragen:

`http://localhost/otrs/nph-genericinterface.pl/Webservice/Test/Resource`

`http://localhost/otrs/nph-genericinterface.pl/Webservice/Test/Resource?Param1=One`

Route mapping: `/Resource/OtherResource/:ID/:Color`

- Gültige Anfragen:

`http://localhost/otrs/nph-genericinterface.pl/Webservice/Test/Resource/OtherResource/1/Red`

`http://localhost/otrs/nph-genericinterface.pl/Webservice/Test/Resource/OtherResource/123/Blue?Param1=One`

In the first example ID = 1 and Color = Red, while in the second ID = 123 and Color = Blue.

- Ungültige Anfragen:

`http://localhost/otrs/nph-genericinterface.pl/Webservice/Test/Resource/1`

`http://localhost/otrs/nph-genericinterface.pl/Webservice/Test/Resource/OtherResource/1`

`http://localhost/otrs/nph-genericinterface.pl/Webservice/Test/Resource/OtherResource/1?Param1=One`

In the first example the part of the path `'/OtherResource'` is missing as well as the `:Color` variable, on the second example just `:Color` variable is missing.

- Valid request methods for Operation '`<OperationName>`':

The HTTP request methods to determine the operation to use together with the route mapping, possible options: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT and TRACE.

Totally different operations can share exactly the same mapping path, but the request method must be unique for each operation, in order to determine correctly the operation to use on each request.

Abbildung 4.108. Web service provider network transport (HTTP::REST)

GenericInterface Transport HTTP::REST for Web Service Webservice two

You are here: > Web Services > Webservice two > Provider Transport HTTP::REST

Actions

Go back to web service

Network transport

Properties

Type: HTTP::REST

★ Maximum message length:

Here you can specify the maximum size (in bytes) of REST messages that OTRS will process.

★ Send Keep-Alive:

This configuration defines if incoming connections should get closed or kept alive.

Save or Save and finish or Cancel

11.4.3.7.2. Webservice-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 share 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.

Normalerweise gibt es zwei Mapping-Konfigurationsabteilungen bei jeder Operation, eine für die ankommenden Daten und eine für die ausgehenden Daten. Sie können verschiedene Mapping-Arten (Backends) für jede Mapping-Richtung auswählen, da deren Konfiguration unabhängig voneinander und unabhängig vom Operations-Backend ist. Die normale und am meisten übliche Praxis ist es, dass die Operation in beiden Fällen die gleiche Mapping-Art nutzt (mit umgekehrter Konfiguration). Die komplette Mapping-Konfiguration wird in einer extra Anzeige, die auf die Mapping-Art drauf ankommt, gemacht.

The operation backend is pre-populated and is not editable. You will see this parameter when you choose the operation on the web service edit screen. The field is only informative.

In the left part of the screen on the action column you have the options: "Go back to web service" (discarding all changes since the last save) and "Delete". If you click on the last one, a dialog will open and ask you if you like to remove the operation. Click on "Delete" button to confirm the removal of the operation and its configuration or "Cancel" to close the delete dialog.

Abbildung 4.109. Web service operation

Change Operation CreateTicket of Web Service Webservice one

You are here: > Web Services > Webservice one > Change operation CreateTicket

Actions

◀ Go back to web service

🗑 Delete

Operation Details

★ Name:
The name is typically used to call up this web service operation from a remote system.

Description:

Mapping for incoming request data: The request data will be processed by this mapping, to transform it to the kind of data OTRS expects.

Operation backend:
This OTRS operation backend module will be called internally to process the request, generating data for the response.

Mapping for outgoing response data: The response data will be processed by this mapping, to transform it to the kind of data the remote system expects.

or or

11.4.3.7.3. Webservice Requester Übertragung

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:

- Verschlüsselung (z. B. utf-8, latin1, iso-8859-1, cp1250, usw.) für die SOAP-Nachricht.
- 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.

Abbildung 4.110. Web service requester network transport (HTTP::SOAP)

GenericInterface Transport HTTP::SOAP for Web Service Webservice one

You are here: > Web Services > Webservice one > Requester Transport HTTP::SOAP

Actions

Network transport
 Properties

Type: HTTP::SOAP

★ Endpoint:
 URI to indicate a specific location for accessing a service.
 e.g. http://local.otrs.com:8000/Webservice/Example

★ Namespace:
 URI to give SOAP methods a context, reducing ambiguities.
 e.g. urn:otrs-com:soap:functions or http://www.otrs.com/GenericInterface/actions

★ Request name scheme:
 Select how SOAP request function wrapper should be constructed.
 'FunctionName' is used as example for actual invoker/operation name.
 'FreeText' is used as example for actual configured value.

★ Response name scheme:
 Select how SOAP response function wrapper should be constructed.
 'FunctionName' is used as example for actual invoker/operation name.
 'FreeText' is used as example for actual configured value.

Encoding:
 The character encoding for the SOAP message contents.
 e.g. utf-8, latin1, iso-8859-1, cp1250, Etc.

SOAPAction:
 Set to "Yes" to send a filled SOAPAction header.
 Set to "No" to send an empty SOAPAction header.

SOAPAction separator:
 Character to use as separator between name space and SOAP method.
 Usually .Net web services uses a "/" as separator.

Authentication:
 The authentication mechanism to access the remote system.
 A "-" value means no authentication.

Proxy Server:
 URI of a proxy server to be used (if needed).
 e.g. http://proxy_hostname:8080

Proxy User:
 The user name to be used to access the proxy server.

Proxy Password:
 The password for the proxy user.

Use SSL Options:
 Show or hide SSL options to connect to the remote system.

Sort options:

 Outbound sort order for xml fields (structure starting below function name wrapper) - see documentation for SOAP transport.

or

In the case of HTTP::REST, this configuration also grows dynamically depending on the configured invokers by adding "Controller mapping for Invoker '<InvokerName>':" and "Valid request command for Invoker '<InvokerName>':" for each invoke. Authentication and SSL options are similar to the ones in HTTP::SOAP

- Host

The host name or IP Address and port of the remote system, if no port is specified, port 80 is used by default.

- Controller-Mapping für Invoker '<InvokerName>':

In this setting a resource path is set. This path must be defined according to the needs of the remote web service and following its definition.

Path can contain variables in the form of '<VariableName>' for each variable name that matches the current data (to be sent), will be replaced by the corresponding data value. This matched variable names and values will be removed from the current data. Depending on the HTTP request command the remaining data could be sent as a JSON string in the request body or as query parameters within the URI.

Beispiele:

Für Daten: Var1 = Eins, Var2 = Zwei, Var3 = Drei und Var4 = Vier.

Controller mapping: /Resource

- After Replacements:

/Ressource

- Verbleibende Daten:

Var1 = Eins, Var2 = Zwei, Var3 = Drei und Var4 = Vier

Controller mapping: /Resource/:Var1

- After Replacements:

/Ressource/Eins

- Verbleibende Daten:

Var2 = Zwei, Var3 = Drei und Var4 = Vier

Controller mapping: /Resource/:Var1?Param1=:Var2&Var3=:Var3

- After Replacements:

/Ressource/Eins?Param1=Zwei&Var3=Drei

- Verbleibende Daten:

Var4 = Vier

- Valid request command for Invoker '<InvokerName>':

This determine the HTTP request method to use, possible options: CONNECT, DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT and TRACE. If no command is selected, Default command is used.

- Standardbefehl

Used as a fall-back for all Invokers without a defined request command.

Abbildung 4.111. Web service provider network transport (HTTP::REST)

GenericInterface Transport HTTP::REST for Web Service Webservice one

You are here: > Web Services > Webservice one > Requester Transport HTTP::REST

Actions

[Go back to web service](#)

Network transport

Properties

Type: HTTP::REST

● Host:
Remote host URL for the REST requests.
 e.g https://www.otrs.com:10745/api/v1.0 (without trailing backslash)

● Controller mapping for Invoker 'InvokerOne':
The controller that the invoker should send requests to. Variables marked by a ':' will get replaced by the data value and passed along with the request. (e.g. /Ticket/TicketID?UserLogin=UserLogin&Password=Password).

Valid request command for Invoker 'InvokerOne':
A specific HTTP command to use for the requests with this Invoker (optional).

Default command:
The default HTTP command to use for the requests.

Authentication:
The authentication mechanism to access the remote system.
 A "*" value means no authentication.

Use SSL Options:
Show or hide SSL options to connect to the remote system.

or or

11.4.3.7.4. Webservice 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). Usually 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. Additionally the list of event triggers and 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-populated and can not be edited. You will see this parameter when you choose the invoker on the web service edit screen. The field is only informative.

Event triggers are events within OTRS such as "TicketCreate", "ArticleSend", etc. These can act as triggers to execute the invoker. Each invoker needs to have at least one event trigger registered, or the invoker will be useless, because it will never be called. The asynchronous property of the event triggers define if the OTRS process will handle the invoker or if it will be delegated to the OTRS Daemon.

Anmerkung

The OTRS Daemon is a separate set of 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 OTRS Daemons using web services can make OTRS slow or non-responsive. Therefore it is highly recommend to use asynchronous event triggers as often as possible.

To add an Event trigger, first select the event family from the first list, then the event name from the second list, then set the asynchronous property (if unchecked means that the event trigger will not be asynchronous) and finally click on the plus button. A new event trigger will be created and it will be listed on the invoker "Event Triggers" list.

To delete an Event trigger, simply locate the event trigger to be deleted in the "Event Triggers" list and click on the trash icon at the end of the row. This will open a dialog that asks you if you are sure to delete the event trigger. Click "Delete" to remove the event trigger from the list, or "Cancel" to close the dialog.

In the left part of the screen on the action column you have the options: "Go back to web service" (discarding all changes since the last save) and "Delete". If you click on the last one, a dialog will emerge and ask you if you like to remove the invoker. Click on the "Delete" button to confirm the removal of the invoker and its configuration or "Cancel" to close the delete dialog.

Abbildung 4.112. Web service invoker

Change Invoker InvokerOne of Web Service Webservice one

You are here: > Web Services > Webservice one > Change invoker InvokerOne

Actions

◀ Go back to web service

🗑 Delete

Invoker Details

★ Name:
The name is typically used to call up an operation of a remote web service.

Description:

Invoker backend:
This OTRS invoker backend module will be called to prepare the data to be sent to the remote system, and to process its response data.

Mapping for outgoing request data:
The data from the invoker of OTRS will be processed by this mapping, to transform it to the kind of data the remote system expects.

Mapping for incoming response data:
The response data will be processed by this mapping, to transform it to the kind of data the invoker of OTRS expects.

Event Triggers:

EVENT	ASYNCHRONOUS	DELETE
TicketCreate	Yes	🗑

This invoker will be triggered by the configured events.

Add Event Trigger: Asynchronous

To add a new event select the event object and event name and click on the "+*" button.
Asynchronous event triggers are handled by the OTRS Scheduler Daemon in background (recommended).
Synchronous event triggers would be processed directly during the web request.

or or

11.4.3.7.5. Webservice Mapping

There are cases where you need to transform the data from one format to another (map or change data structure), because normally a web service is used to interact with a Remote System, that is highly probable that is not another OTRS system and / or could not understand the OTRS data structures and values. In these cases some or all values have to be changed, and sometimes even the names of the values (keys) or even the complete structure, in order to match with the expected data on the other end. To accomplish this task the Generic Interface Mapping Layer exists.

Each Remote System has its 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 its associated value is deleted too. When this is applied to the value, only the value is deleted, keeping the key, that now will be associated to an empty value.
- MapTo (use provided key or value as default): all keys and / or values without a defined map rule, will use this as default, when you select this option a new text field will appear to set this default.

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

- 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 it is also possible to define rules for each value to be mapped with the same options as for the key map (Exact value and Regular expression). You can add as many values to map as needed, and if you want to delete one of them, just click on the "-" button for each mapping value row.

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.

Abbildung 4.113. Web service mapping

GenericInterface Mapping Simple for Web Service Webservice one

You are here: Web Services > Webservice one > Operation CreateTicket > Simple Mapping for Outgoing Data

Actions: Go back to operation

Mapping Simple

Default rule for unmapped keys: This rule will apply for all keys with no mapping rule.

Default rule for unmapped values: This rule will apply for all values with no mapping rule.

New key map:

▼ Mapping for Key KeyNew

Key mapping: ★ Map key: matching the: ★ to new key:

Value mapping: ★ Map value: matching the: ★ to new value:

★ Map value: matching the: ★ to new value:

New value map:

or or

11.5. Webdienst-Befehlszeilenoberfläche

The `bin/otrs.Console.pl Admin::WebService::*` commands were developed in order to create basic, but fast and powerful tools to work with web service configurations. They give you the ability to perform the following actions:

- Add: Um einen Webservice mit einer YAML-Datei als Konfigurationsquelle zu erstellen.
- Update: to change an existing web service, the configuration can be changed using a different or modified YAML file.
- Dump: to save the current web service configuration to a file.
- 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.

Beispiel: Eine neue Webservice-Konfiguration erstellen:

```
shell> bin/otrs.Console.pl Admin::WebService::Add --name <webservice_name> --source-path /  
path/to/yaml/file
```

11.6. Webservice Konfiguration

From its design the web services were conceived to be portable from one OTRS system to another, e.g. from a test or development environment to a production system. Therefore it was needed to have an easy way to extract the web service configuration from the database, and import it to another. To accomplish this task the Generic Interface uses YAML files as the web services configuration basis.

Why YAML? YAML is a markup language designed to be human friendly to read and write (it is easier to understand than JSON), it does not have some of the limitations of XML like numeric tags, it is open, standardized, and is complete enough to store the whole web service configuration.

Anmerkung

To learn more about YAML please visit <http://www.yaml.org/>.

Im folgenden finden Sie eine Webservice-Konfigurationsdatei im 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
```

11.6.1. Konfigurationsdetails

11.6.1.1. Allgemein

- Beschreibung: ein kurzer Text, der den Webservice beschreibt.
- RemoteSystem: eine kurze Beschreibung des Remote-Systems.

- Debugger: ein Behälter für die Debugger-Einstellungen.
- Provider: ein Behälter für die Anbieter-Einstellungen.
- Requester: ein Behälter für die Anforderer-Einstellungen.

11.6.1.2. Debugger

- DebugThreshold: the debugger level.

mögliche Werte

- debug: alle Protokolle werden in der Datenbank gespeichert.
- info: info, notice und error level - Protokolle werden in der Datenbank gespeichert.
- notice: notice und error level - Protokolle werden in der Datenbank gespeichert.
- error: nur error level - Protokolle werden in der Datenbank gespeichert.

11.6.1.3. Provider

- Operation: ein Behälter für alle Operations-Einstellungen.
- Transport: ein Behälter für Provider-Netzwerk-Transport-Einstellungen.

11.6.1.3.1. Operation

- <OperationName>: eindeutiger Name für die Operation, eigener Container für eigene Operationseinstellungen (Kardinalität 0 ..n, aber nicht mehrfach).

11.6.1.3.1.1. <Operationsname>

Dieser Abschnitt basiert auf Operationen vom Typ "Test::Test" andere Operationen können unterschiedliche Einstellungen beinhalten.

- Beschreibung: ein kurzer Text, der die Operation beschreibt.
- MappingInbound: Ein Container für die Mapping-Einstellung für eingehende Anfragen.
- MappingOutbound: Ein Container für die Mapping-Einstellung für ausgehende Anfragen.
- Type: Das Operation-Backend, im Format Controller::Operation.

11.6.1.3.1.1.1. eingehendes Mapping

Dieser Abschnitt basiert auf Mappings vom Typ "Simple". Andere Mappings können eventuell mehr oder andere Einstellungen beinhalten.

- Config: Ein Container für diese Mapping-Einstellungen.
- Type: Das Mapping-Backend.

11.6.1.3.1.1.1.1. Konfiguration

- KeyMapDefault: Ein Container für alle nicht gemappten Key-Einstellungen.
- ValueMapDefault: Ein Container für alle nicht gemappten Wert-Einstellungen.
- KeyMapExact: Ein Container für alle exakten Key-Mappings (Kardinalität 0 .. 1).
- KeyMapRegEx: Ein Container für alle regulären Regular Expression Key-Mappings (Kardinalität 0 .. 1).
- ValueMap: Ein Container für alle Werte-Mappings (Kardinalität 0 .. 1).

11.6.1.3.1.1.1.1. KeyMapDefault

- MapTo: Der neue Wert welcher benutzt werden soll (nur anwendbar wenn der MapType auf MapTo gesetzt ist).
- MapType: Die Regel für das Mapping.

mögliche Werte

- Keep: unverändert lassen.
- Ignore: verwerfen.
- MapTo: Ändern auf den MapTo-Wert.

11.6.1.3.1.1.1.2. ValueMapDefault

Ähnlich zu KeyMapDefault.

11.6.1.3.1.1.1.3. KeyMapExact

- <alterKey>: <neuerKey> (Kardinalität 0 .. n aber nicht mehrfach).

11.6.1.3.1.1.1.4. KeyMapRegEx

- <oldkey(RegEx)>: <newkey> (Kardinalität 0 .. n, aber nicht mehrfach).

11.6.1.3.1.1.1.5. ValueMap

- <newkey>: Ein Container für das Werte-Mapping für diesen neuen Key (Kardinalität hängt von den neuen Keys aus KeyMapExact und KeyMapRegEx ab).

11.6.1.3.1.1.1.5.1. <newkey>

- ValueMapExact: Ein Container für alle exakten Werte-Mappings (Kardinalität 0 .. 1).
- ValueMapRegEx: Ein Container für alle Regular Expression Werte-Mappings (Kardinalität 0 .. 1).

11.6.1.3.1.1.1.5.1.1. ValueMapExact

- <oldvalue>: <newvalue> (Kardinalität 0 .. n, aber nicht mehrfach).

11.6.1.3.1.1.1.5.1.2. ValueMapRegEx

- <oldvalue(RegEx)>: <newvalue> (Kardinalität 0 .. n, aber nicht mehrfach).

11.6.1.3.1.1.2. ausgehendes Mapping

Das gleiche wie beim eingehenden Mapping

11.6.1.3.1.1.3. Transport

Dieser Abschnitt basiert auf Provider-Übertragungsart HTTP::SOAP, andere Übertragungsarten können unterschiedliche Einstellungen beinhalten.

- Config: Ein Container für die spezifischen Netzwerk-Transport-Einstellungen.
- Type: Netzwerk-Transport-Backend des Providers.

11.6.1.3.1.1.3.1. Konfiguration

- MaxLength: Die maximale Länge in Bytes die in SOAP-Nachrichten von OTRS gelesen werden.
- NameSpace: eine URI die eine Zusammenfassung aller möglichen Operationen liefert, die zu diesem Webservice gehören.

11.6.1.4. Requester

- Invoker: ein Container für jede 'Invoker'-Einstellung.
- Transport: ein Container für Netzwerk-Transport-Einstellungen für Requester.

11.6.1.4.1. Invoker

- <InvokerName>: eindeutiger Name für den Invoker, Container für eigene Invoker-Einstellungen (Kardinalität 0..n, aber nicht mehrfach).

11.6.1.4.1.1. <InvokerName>

Dieser Abschnitt basiert auf Invoker vom Typ "Test::Test" andere Operationen können unterschiedliche Einstellungen beinhalten.

- Description: a short text that describes the invoker.
- Events: ein Container für Eventtrigger-Einstellungen, die keinen Namen besitzen.
- MappingInbound: Ein Container für die Mapping-Einstellung für eingehende Anfragen.
- MappingOutbound: Ein Container für die Mapping-Einstellung für ausgehende Anfragen.
- Type: Das Invoker-Backend, im Format Controller::Invoker

11.6.1.4.1.1.1. Ereignisse

- *List Element*: (cardinality 0 .. n).
- Asynchronous: to set if the invoker execution will be delegated to the OTRS Daemon.

mögliche Werte

- 0: not handled by the OTRS Daemon.
- 1: handled by the OTRS Daemon.
- Event: Der Name des Event-Triggerers.

Mögliche Werte (für TicketEvents)

- 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

Mögliche Werte (für Artikel-Events)

- ArticleCreate
- ArticleFreeTextUpdate
- ArticleUpdate
- ArticleSend
- ArticleBounce
- ArticleAgentNotification
- ArticleCustomerNotification
- ArticleAutoResponse
- ArticleFlagSet
- ArticleFlagDelete
- ArticleAgentNotification
- ArticleCustomerNotification

11.6.1.4.1.1.2. eingehendes Mapping

Das gleiche wie bei der Operation MappingInbound

11.6.1.4.1.1.3. ausgehendes Mapping

Das gleiche wie bei der Operation MappingInbound

11.6.1.4.1.1.4. Transport

Dieser Abschnitt basiert auf Requester-Übertragungsart HTTP::SOAP, andere Übertragungsarten können unterschiedliche Einstellungen beinhalten.

- Config: Ein Container für die spezifischen Netzwerk-Transport-Einstellungen.
- Type: Netzwerk-Transport-Backend des Requesters.

11.6.1.4.1.1.4.1. Konfiguration

- Authentifizierung: ein Behälter für die Authentifizierungs-Einstellungen.
- 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).

mögliche Werte

- Yes: to send a filled SOAPAction header.
- Nein: um einen leeren SOAPAction-Header zu senden.
- SOAPActionSeparator: Für das setzen eines <Separator> eines SOAPAction Headers.

mögliche Werte

- '/': genutzt für .net-Webservices.
- '#': genutzt für alle anderen Webservices.

11.6.1.4.1.1.4.1.1. Authentifizierung

- User: der vertrauliche Benutzername, der Zugang zu dem entfernten Webservice hat.
- Passwort: das Passwort für den vertraulichen Benutzer im Klartext.
- Type: der Typ der Authentifizierung.

11.7. Verbinder

A Connector is in essence a set of actions that are either called Operations if OTRS acts as a web service provider or Invokers if OTRS acts as a web service requester. But it can also include special Mappings or Transports.

One Connector can either have only Operations, Only Invokers or both. A connector can even use parts of other connectors like the Mappings or Transports if they are not to specific for the Connector that is trying to implement them.

In other words a Connector is not limited to just the Controller layer but it can be extended to Data Mapping or Network Transport layers if needed.

Due to the modular design of the Generic Interface a Connector can be seen as a plug-in; this means that by adding Connectors the capabilities of the generic interface can be extended using: OTRS Feature add ons, OTRS Custom modules, 3rd Party modules, and so on.

11.7.1. gebündelte Verbinder

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

- Sitzung
- Ticket

11.7.1.1. Session-Connector

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

Liefert:

- Operationen:
 - SessionCreate

11.7.1.1.1. Operationen

11.7.1.1.1.1. SessionCreate

Creates a 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 it is necessary that the operation implements authentication by SessionID. All the rest of the bundled operations are capable of accepting a valid SessionID as an authentication method.

Mögliche Attribute:

```
<SessionCreate>
<!--You have a MANDATORY CHOICE of the next 2 items at this level-->
<!--Optional:-->
<UserLogin>?</UserLogin>
<!--Optional:-->
<CustomerUserLogin>?</CustomerUserLogin>
<!--Optional:-->
<Password>?</Password>
</SessionCreate>
```

11.7.1.2. Ticket Connector

This connector supplies the basic functionality to interact with tickets.

Liefert:

- Operationen:
 - TicketCreate
 - TicketUpdate

- TicketGet
- TicketSearch

11.7.1.2.1. Operationen

11.7.1.2.1.1. TicketCreate

Provides an interface to create a ticket in OTRS. A ticket must contain an Article and can contain several attachments, all defined Dynamic Fields can be also set on TicketCreate operation.

Mögliche Attribute:

```

<TicketCreate>
  <!--You have a MANDATORY CHOICE of the next 3 items at this level-->
  <!--Optional:-->
  <UserLogin?</UserLogin>
  <!--Optional:-->
  <CustomerUserLogin?</CustomerUserLogin>
  <!--Optional:-->
  <SessionID?</SessionID>
  <!--Optional:-->
  <Password?</Password>
  <Ticket>
    <Title?</Title>
    <!--You have a MANDATORY CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <QueueID?</QueueID>
    <!--Optional:-->
    <Queue?</Queue>
    <!--You have a CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <TypeID?</TypeID>
    <!--Optional:-->
    <Type?</Type>
    <!--You have a CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <ServiceID?</ServiceID>
    <!--Optional:-->
    <Service?</Service>
    <!--You have a CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <SLAID?</SLAID>
    <!--Optional:-->
    <SLA?</SLA>
    <!--You have a MANDATORY CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <StateID?</StateID>
    <!--Optional:-->
    <State?</State>
    <!--You have a MANDATORY CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <PriorityID?</PriorityID>
    <!--Optional:-->
    <Priority?</Priority>
    <!--You have a CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <OwnerID?</OwnerID>
    <!--Optional:-->
    <Owner?</Owner>
    <!--You have a CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <ResponsibleID?</ResponsibleID>
    <!--Optional:-->
    <Responsible?</Responsible>
    <CustomerUser?</CustomerUser>
  
```

```

<!--Optional:-->
<CustomerID?</CustomerID>
<!--Optional:-->
<PendingTime>
  <!--You have a CHOICE of the next and the other 5 items at this level-->
  <Diff?</Diff>
  <Year?</Year>
  <Month?</Month>
  <Day?</Day>
  <Hour?</Hour>
  <Minute?</Minute>
</PendingTime>
</Ticket>
<Article>
  <!--You have a CHOICE of the next 2 items at this level-->
  <!--Optional:-->
  <ArticleTypeID?</ArticleTypeID>
  <!--Optional:-->
  <ArticleType?</ArticleType>
  <!--You have a CHOICE of the next 2 items at this level-->
  <!--Optional:-->
  <SenderTypeID?</SenderTypeID>
  <!--Optional:-->
  <SenderType?</SenderType>
  <!--Optional:-->
  <From?</From>
  <Subject?</Subject>
  <Body?</Body>
  <!--You have a CHOICE of the next 2 items at this level-->
  <!--Optional:-->
  <ContentType?</ContentType>
  <Charset?</Charset>
  <MimeType?</MimeType>
  <!--Optional:-->
  <HistoryType?</HistoryType>
  <!--Optional:-->
  <HistoryComment?</HistoryComment>
  <!--Optional:-->
  <AutoResponseType?</AutoResponseType>
  <!--Optional:-->
  <TimeUnit?</TimeUnit>
  <!--Optional:-->
  <NoAgentNotify?</NoAgentNotify>
  <!--Zero or more repetitions:-->
  <ForceNotificationToUserID?</ForceNotificationToUserID>
  <!--Zero or more repetitions:-->
  <ExcludeNotificationToUserID?</ExcludeNotificationToUserID>
  <!--Zero or more repetitions:-->
  <ExcludeMuteNotificationToUserID?</ExcludeMuteNotificationToUserID>
</Article>
<!--Zero or more repetitions:-->
<DynamicField>
  <Name?</Name>
  <!--1 or more repetitions:-->
  <Value?</Value>
</DynamicField>
<!--Zero or more repetitions:-->
<Attachment>
  <Content>cid:61886944659</Content>
  <ContentType?</ContentType>
  <Filename?</Filename>
</Attachment>
</TicketCreate>

```

11.7.1.2.1.2. TicketUpdate

TicketUpdate operation adds the capability to modify attributes from an existing ticket or to add a new article, including attachments and all defined dynamic fields for the ticket and the new article.

Anmerkung

It is not necessary to create a new article to modify a ticket attribute.

Mögliche Attribute:

```

<TicketUpdate>
  <!--You have a MANDATORY CHOICE of the next 3 items at this level-->
  <!--Optional:-->
  <UserLogin?</UserLogin>
  <!--Optional:-->
  <CustomerUserLogin?</CustomerUserLogin>
  <!--Optional:-->
  <SessionID?</SessionID>
  <!--Optional:-->
  <Password?</Password>
  <!--You have a CHOICE of the next 2 items at this level-->
  <TicketID?</TicketID>
  <TicketNumber?</TicketNumber>
  <!--Optional:-->
  <Ticket>
    <!--Optional:-->
    <Title?</Title>
    <!--You have a CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <QueueID?</QueueID>
    <!--Optional:-->
    <Queue?</Queue>
    <!--You have a CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <TypeID?</TypeID>
    <!--Optional:-->
    <Type?</Type>
    <!--You have a CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <ServiceID?</ServiceID>
    <!--Optional:-->
    <Service?</Service>
    <!--You have a CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <SLAID?</SLAID>
    <!--Optional:-->
    <SLA?</SLA>
    <!--You have a CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <StateID?</StateID>
    <!--Optional:-->
    <State?</State>
    <!--You have a CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <PriorityID?</PriorityID>
    <!--Optional:-->
    <Priority?</Priority>
    <!--You have a CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <OwnerID?</OwnerID>
    <!--Optional:-->
    <Owner?</Owner>
    <!--You have a CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <ResponsibleID?</ResponsibleID>
    <!--Optional:-->
    <Responsible?</Responsible>
    <!--Optional:-->
    <CustomerUser?</CustomerUser>
    <!--Optional:-->
    <CustomerID?</CustomerID>
    <!--Optional:-->
    <PendingTime>

```



```

        <!--You have a CHOICE of the next and the other 5 items at this level-->
        <Diff?</Diff>
        <Year?</Year>
        <Month?</Month>
        <Day?</Day>
        <Hour?</Hour>
        <Minute?</Minute>
    </PendingTime>
</Ticket>
<!--Optional:-->
<Article>
    <!--You have a CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <ArticleTypeID?</ArticleTypeID>
    <!--Optional:-->
    <ArticleType?</ArticleType>
    <!--You have a CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <SenderTypeID?</SenderTypeID>
    <!--Optional:-->
    <SenderType?</SenderType>
    <!--Optional:-->
    <From?</From>
    <Subject?</Subject>
    <Body?</Body>
    <!--You have a CHOICE of the next 2 items at this level-->
    <!--Optional:-->
    <ContentType?</ContentType>
    <Charset?</Charset>
    <MimeType?</MimeType>
    <!--Optional:-->
    <HistoryType?</HistoryType>
    <!--Optional:-->
    <HistoryComment?</HistoryComment>
    <!--Optional:-->
    <AutoResponseType?</AutoResponseType>
    <!--Optional:-->
    <TimeUnit?</TimeUnit>
    <!--Optional:-->
    <NoAgentNotify?</NoAgentNotify>
    <!--Zero or more repetitions:-->
    <ForceNotificationToUserID?</ForceNotificationToUserID>
    <!--Zero or more repetitions:-->
    <ExcludeNotificationToUserID?</ExcludeNotificationToUserID>
    <!--Zero or more repetitions:-->
    <ExcludeMuteNotificationToUserID?</ExcludeMuteNotificationToUserID>
</Article>
<!--Zero or more repetitions:-->
<DynamicField>
    <Name?</Name>
    <!--1 or more repetitions:-->
    <Value?</Value>
</DynamicField>
<!--Zero or more repetitions:-->
<Attachment>
    <Content>cid:166861569966</Content>
    <ContentType?</ContentType>
    <Filename?</Filename>
</Attachment>
</TicketUpdate>

```

11.7.1.2.1.3. TicketGet

This operation is used to get all the attributes of a ticket including the dynamic fields, all articles and all of the attachments that belong to the ticket.

Mögliche Attribute:

```

<TicketGet>
<!--You have a MANDATORY CHOICE of the next 3 items at this level-->
<!--Optional:-->
<UserLogin?></UserLogin>
<!--Optional:-->
<CustomerUserLogin?></CustomerUserLogin>
<!--Optional:-->
<SessionID?></SessionID>
<!--Optional:-->
<Password?></Password>
<!--1 or more repetitions:-->
<TicketID?></TicketID>
<!--Optional:-->
<DynamicFields?></DynamicFields>
<!--Optional:-->
<Extended?></Extended>
<!--Optional:-->
<AllArticles?></AllArticles>
<!--Optional:-->
<ArticleSenderType?></ArticleSenderType>
<!--Optional:-->
<ArticleOrder?></ArticleOrder>
<!--Optional:-->
<ArticleLimit?></ArticleLimit>
<!--Optional:-->
<Attachments?></Attachments>
</TicketGet>

```

11.7.1.2.1.4. TicketSearch

Die Operation TicketSearch liefert eine Liste mit TicketIDs die auf die eingegebenen Suchkriterien passen.

Mögliche Attribute:

```

<TicketSearch>
  <!--You have a MANDATORY CHOICE of the next 3 items at this level-->
  <!--Optional:-->
  <UserLogin?></UserLogin>
  <!--Optional:-->
  <CustomerUserLogin?></CustomerUserLogin>
  <!--Optional:-->
  <SessionID?></SessionID>
  <!--Optional:-->
  <Password?></Password>
  <!--Optional:-->
  <Limit?></Limit>
  <!--Zero or more repetitions:-->
  <TicketNumber?></TicketNumber>
  <!--Zero or more repetitions:-->
  <Title?></Title>
  <!--Zero or more repetitions:-->
  <Queues?></Queues>
  <!--Zero or more repetitions:-->
  <QueueIDs?></QueueIDs>
  <!--Optional:-->
  <UseSubQueues?></UseSubQueues>
  <!--Zero or more repetitions:-->
  <Types?></Types>
  <!--Zero or more repetitions:-->
  <TypeID?></TypeID>
  <!--Zero or more repetitions:-->
  <States?></States>
  <!--Zero or more repetitions:-->
  <StateIDs?></StateIDs>
  <!--Zero or more repetitions:-->
  <StateType?></StateType>
  <!--Zero or more repetitions:-->

```

```

<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:-->
<TicketLastChangeTimeOlderMinutes?></TicketLastChangeTimeOlderMinutes>
<!--Optional:-->
<TicketLastChangeTimeNewerMinutes?></TicketLastChangeTimeNewerMinutes>
<!--Optional:-->
<TicketLastChangeTimeNewerDate?></TicketLastChangeTimeNewerDate>
<!--Optional:-->
<TicketLastChangeTimeOlderDate?></TicketLastChangeTimeOlderDate>
<!--Optional:-->
<TicketChangeTimeOlderMinutes?></TicketChangeTimeOlderMinutes>
<!--Optional:-->
<TicketChangeTimeNewerMinutes?></TicketChangeTimeNewerMinutes>
<!--Optional:-->
<TicketChangeTimeNewerDate?></TicketChangeTimeNewerDate>
<!--Optional:-->
<TicketChangeTimeOlderDate?></TicketChangeTimeOlderDate>
<!--Optional:-->
<TicketCloseTimeOlderMinutes?></TicketCloseTimeOlderMinutes>
<!--Optional:-->
<TicketCloseTimeNewerMinutes?></TicketCloseTimeNewerMinutes>
<!--Optional:-->
<TicketCloseTimeNewerDate?></TicketCloseTimeNewerDate>
<!--Optional:-->
<TicketCloseTimeOlderDate?></TicketCloseTimeOlderDate>
<!--Optional:-->
<TicketPendingTimeOlderMinutes?></TicketPendingTimeOlderMinutes>
<!--Optional:-->
<TicketPendingTimeNewerMinutes?></TicketPendingTimeNewerMinutes>
<!--Optional:-->
<TicketPendingTimeNewerDate?></TicketPendingTimeNewerDate>
<!--Optional:-->
<TicketPendingTimeOlderDate?></TicketPendingTimeOlderDate>
<!--Optional:-->
<TicketEscalationTimeOlderMinutes?></TicketEscalationTimeOlderMinutes>
<!--Optional:-->
<TTicketEscalationTimeNewerMinutes?></TTicketEscalationTimeNewerMinutes>
<!--Optional:-->
<TicketEscalationTimeNewerDate?></TicketEscalationTimeNewerDate>
<!--Optional:-->
<TicketEscalationTimeOlderDate?></TicketEscalationTimeOlderDate>
<!--Optional:-->
<ArchiveFlags?></ArchiveFlags>
<!--Zero or more repetitions:-->
<OrderBy?></OrderBy>
<!--Zero or more repetitions:-->
<SortBy?></SortBy>

```

```
<!--Zero or more repetitions:-->
<CustomerUserID?</CustomerUserID>
</TicketSearch>
```

11.7.2. Beispiele:

11.7.2.1. Webservice Konfiguration

The following is a basic but complete web service configuration file in YAML format to use all the Ticket Connector operations with the SOAP network transport. In order to use it in OTRS you need to copy the content, save it into a file and call it `GenericTicketConnectorSOAP.yml`, and import it into OTRS in the Web Services screen in the Admin panel by clicking in the "Add web service" button from the overview screen and then clicking in the "Import web service" button in the add screen.

```
---
Debugger:
  DebugThreshold: debug
  TestMode: 0
Description: Ticket Connector SOAP Sample
FrameworkVersion: 3.4.x git
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: Retrieves 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: ''
```

Similar example can be done for the REST network transport, REST web services uses HTTP operations such as "POST", "GET", "PUT", "PATCH" etc. This operations in conjunction with a URI path called resource defines a OTRS Generic Interface Operation or Invoker (depending on the communication way).

The following example uses /Session resource for SessionCreate, /Ticket resource for TicketSearch and TicketCreate and resource /Ticket/{TicketID} for TicketGet and TicketUpdate (Where {TicketID} is the actual TicketID value of a ticket e.g. /Ticket/123). In order to use it in OTRS you need to copy the content, save it into a file and call it GenericTicketConnectorREST.yml, and import it into OTRS in the Web Services screen in the Admin panel by clicking in the "Add web service" button from the overview screen and then clicking in the "Import web service" button in the add screen.

```

---
Debugger:
  DebugThreshold: debug
  TestMode: '0'
Description: Ticket Connector REST Sample
FrameworkVersion: 3.4.x git
Provider:
  Operation:
    SessionCreate:
      Description: Creates a Session
      MappingInbound: {}
      MappingOutbound: {}
      Type: Session::SessionCreate
    TicketCreate:
      Description: Creates a Ticket
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketCreate
    TicketGet:
      Description: Retrieves Ticket data
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketGet
    TicketSearch:
      Description: Search for Tickets
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketSearch
    TicketUpdate:
      Description: Updates a Ticket
      MappingInbound: {}
      MappingOutbound: {}
      Type: Ticket::TicketUpdate
  Transport:
    Config:
      KeepAlive: ''
      MaxLength: '100000000'
      RouteOperationMapping:
        SessionCreate:
          RequestMethod:
            - POST
          Route: /Session
        TicketCreate:
          RequestMethod:
            - POST
          Route: /Ticket
        TicketGet:
          RequestMethod:
            - GET
          Route: /Ticket/:TicketID
        TicketSearch:
          RequestMethod:
            - GET
          Route: /Ticket
        TicketUpdate:
          RequestMethod:
            - PATCH
          Route: /Ticket/:TicketID
      Type: HTTP::REST
    RemoteSystem: ''
  Requester:

```

```
Transport:  
Type: ''
```

11.7.2.2. Perl SOAP Requester

The following code is a Perl script that can connect to OTRS via the generic interface. In order to perform the operations provided by the Ticket Connector, it uses two Perl CPAN modules SOAP::Lite and Data::Dumper. Please make sure that your environment is capable to use these modules before you try to run the script.

```
#!/usr/bin/perl -w  
# --  
# otrs.SOAPRequest.pl - sample to send a SOAP request to OTRS Generic Interface Ticket  
# Connector  
# Copyright (C) 2001-2016 OTRS AG, http://otrs.com/  
# --  
# This program is free software; you can redistribute it and/or modify  
# it under the terms of the GNU AFFERO General Public License as published by  
# the Free Software Foundation; either version 3 of the License, or  
# any later version.  
#  
# This program is distributed in the hope that it will be useful,  
# but WITHOUT ANY WARRANTY; without even the implied warranty of  
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the  
# GNU General Public License for more details.  
#  
# You should have received a copy of the GNU Affero General Public License  
# along with this program; if not, write to the Free Software  
# Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA  
# or see http://www.gnu.org/licenses/agpl.txt.  
# --  
  
use strict;  
use warnings;  
  
# use ../ as lib location  
use File::Basename;  
use FindBin qw($RealBin);  
use lib dirname($RealBin);  
  
use SOAP::Lite;  
use Data::Dumper;  
  
# ---  
# Variables to be defined.  
  
# this is the URL for the web service  
# the format is  
# <HTTP_TYPE>:://<OTRS_FQDN>/nph-genericinterface.pl/Webservice/<WEB_SERVICE_NAME>  
# or  
# <HTTP_TYPE>:://<OTRS_FQDN>/nph-genericinterface.pl/WebserviceID/<WEB_SERVICE_ID>  
my $URL = 'http://localhost/otrs/nph-genericinterface.pl/Webservice/GenericTicketConnector';  
  
# this name space should match the specified name space in the SOAP transport for the web  
# service.  
my $NameSpace = 'http://www.otrs.org/TicketConnector/';  
  
# this is operation to execute, it could be TicketCreate, TicketUpdate, TicketGet,  
# TicketSearch  
# or SessionCreate. and they must to be defined in the web service.  
my $Operation = 'TicketCreate';  
  
# this variable is used to store all the parameters to be included on a request in XML  
# format. Each  
# operation has a determined set of mandatory and non mandatory parameters to work  
# correctly. Please  
# check the OTRS Admin Manual in order to get a complete list of parameters.
```

```

my $XMLData = '
<UserLogin>some user login</UserLogin>
<Password>some password</Password>
<Ticket>
  <Title>some title</Title>
  <CustomerUser>some customer user login</CustomerUser>
  <Queue>some queue</Queue>
  <State>some state</State>
  <Priority>some priority</Priority>
</Ticket>
<Article>
  <Subject>some subject</Subject>
  <Body>some body</Body>
  <ContentType>text/plain; charset=utf8</ContentType>
</Article>
';

# ---

# create a SOAP::Lite data structure from the provided XML data structure.
my $SOAPData = SOAP::Data
  ->type( 'xml' => $XMLData );

my $SOAPObject = SOAP::Lite
  ->uri($NameSpace)
  ->proxy($URL)
  ->$Operation($SOAPData);

# check for a fault in the soap code.
if ( $SOAPObject->fault ) {
  print $SOAPObject->faultcode, " ", $SOAPObject->faultstring, "\n";
}

# otherwise print the results.
else {

  # get the XML response part from the SOAP message.
  my $XMLResponse = $SOAPObject->context()->transport()->proxy()->http_response()-
>content();

  # deserialize response (convert it into a perl structure).
  my $Deserialized = eval {
    SOAP::Deserializer->deserialize($XMLResponse);
  };

  # remove all the headers and other not needed parts of the SOAP message.
  my $Body = $Deserialized->body();

  # just output relevant data and no the operation name key (like TicketCreateResponse).
  for my $ResponseKey ( keys %{$Body} ) {
    print Dumper( $Body->{$ResponseKey} );
  }
}

```

11.7.2.3. Perl REST Requester

The following code is a Perl script that can connect to OTRS via the generic interface. In order to perform the operations provided by the Ticket Connector, it uses three Perl CPAN modules JSON, REST::Client and Data::Dumper. Please make sure that your environment is capable to use these modules before you try to run the script.

```

#!/usr/bin/perl
# --
# otrs.RESTRequest.pl - sample to send a REST request to OTRS Generic Interface Ticket
Connector
# Copyright (C) 2001-2016 OTRS AG, http://otrs.com/
# --

```



```

# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU AFFERO General Public License as published by
# the Free Software Foundation; either version 3 of the License, or
# any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU Affero General Public License
# along with this program; if not, write to the Free Software
# Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
# or see http://www.gnu.org/licenses/agpl.txt.
# --

use strict;
use warnings;

## nofilter(TidyAll::Plugin::OTRS::Perl::Dumper)

# use ../ as lib location
use File::Basename;
use FindBin qw($RealBin);
use lib dirname($RealBin);

use JSON;
use REST::Client;

# ---
# Variables to be defined

# This is the HOST for the web service the format is:
# <HTTP_TYPE>:://<OTRS_FQDN>/nph-genericinterface.pl
my $Host = 'http://localhost/otrs/nph-genericinterface.pl';

my $RestClient = REST::Client->new(
    {
        host => $Host,
    }
);

# This is the Controller and Request the format is:
# /Webservice/<WEB_SERVICE_NAME>/<RESOURCE>/<REQUEST_VALUE>
# or
# /WebserviceID/<WEB_SERVICE_ID>/<RESOURCE>/<REQUEST_VALUE>
# This example will retrieve the Ticket with the TicketID = 1 (<REQUEST_VALUE>)
my $ControllerAndRequest = '/Webservice/GenericTicketConnectorREST/Ticket/1';

my $Params = {
    UserLogin    => "some user login",      # to be filled with valid agent login
    Password     => "some user password",  # to be filled with valid agent password
    DynamicFields => 1,                    # optional, if set to 1,
                                                # ticket dynamic fields included in response
    AllArticles  => 1,                    # optional, if set to 1,
                                                # all ticket articles are included in response
                                                # more options to be found in
        # /Kernel/GenericInterface/Operation/Ticket/TicketGet.pm's
        # Run() subroutine documentation.
};

my @RequestParam;

# As sample web service configuration for TicketGet uses HTTP method GET all other
# parameters needs
# to be sent as URI query parameters

# ----
# For GET method
my $QueryParams = $RestClient->buildQuery( %{ $Params } );

$ControllerAndRequest .= $QueryParams;

```

```
# The @RequestParam array on position 0 holds controller and request
@RequestParam = ($ControllerAndRequest);

$RestClient->GET(@RequestParam);
# ----

# # ----
# # For POST method
# my $JSONParams = encode_json $Params;

# # The @RequestParam array on position 0 holds controller and request
# # on position 1 it holds the JSON data string that gets posted
# @RequestParam = (
#   $ControllerAndRequest,
#   $JSONParams
# );

# $RestClient->POST(@RequestParam);
# # ----

# If the host isn't reachable, wrong configured or couldn't serve the requested page:
my $ResponseCode = $RestClient->responseCode();
if ( $ResponseCode ne '200' ) {
    print "Request failed, response code was: $ResponseCode\n";
    exit;
}

# If the request was answered correctly, we receive a JSON string here.
my $ResponseContent = $RestClient->responseContent();

my $Data = decode_json $ResponseContent;

# Just to print out the returned Data structure:
use Data::Dumper;
print "Response was:\n";
print Dumper($Data);
```

11.7.2.4. cURL Beispiel für REST Anfragen.

Given the above example on a REST configuration for Generic Ticket Connector we have that:

Für Ticket-Erstellung: benutzen der POST-Methode am /Ticket-Pfad.

Für Ticket-Suche: benutzen der GET-Methode am /Ticket-Pfad.

Für Ticket-Update: benutzen der PATCH-Methode am /Ticket/{TicketID}-Pfad (in der {TicketID} eine Vorlage in der Übertragungskonfiguration ist)

For Ticket Get: use GET method on /Ticket/{TicketID} path (where {TicketID} is a template represented by :TicketID in the transport configuration)

11.7.2.4.1. Neues Ticket erstellen

cURL Kommando:

```
shell> curl "http://localhost/otrs/nph-genericinterface.pl/Webservice/
GenericTicketConnectorREST/Ticket?UserLogin=agent&Password=123" -H "Content-Type:
application/json" -d "{\"Ticket\":{\"Title\":\"REST Create Test\", \"Type\": \"Unclassified
\", \"Queue\":\"Raw\", \"State\":\"open\", \"Priority\":\"3 normal\", \"CustomerUser\":
\"customer\"}, \"Article\":{\"Subject\":\"Rest Create Test\", \"Body\":\"This is only a test
\", \"ContentType\":\"text/plain; charset=utf8\"}}" -X POST
```

Antwort:

```
{
  "ArticleID":5484,
  "TicketNumber":"1001936",
  "TicketID":"1686"
}
```

11.7.2.4.2. Get Ticket-Eigenschaften

cURL Kommando:

```
curl "http://localhost/otrs/nph-genericinterface.pl/Webservice/GenericTicketConnectorREST/Ticket/1686?UserLogin=agent&Password=123"
```

Antwort:

```
{
  "Ticket": [
    {
      "Age": 777,
      "PriorityID": 3,
      "ServiceID": "",
      "Type": "Unclassified",
      "Responsible": "root@localhost",
      "StateID": 4,
      "ResponsibleID": 1,
      "ChangeBy": 2,
      "EscalationTime": 0,
      "Changed": "2014-06-30 19:08:14",
      "OwnerID": 2,
      "RealTillTimeNotUsed": 0,
      "GroupID": 1,
      "Owner": "agent",
      "CustomerID": "OTRS",
      "TypeID": 1,
      "Created": "2014-06-30 19:08:12",
      "Priority": "3 normal",
      "UntilTime": 0,
      "EscalationUpdateTime": 0,
      "QueueID": 2,
      "Queue": "Raw",
      "State": "open",
      "Title": "REST Create Test",
      "CreateBy": 2,
      "TicketID": 1686,
      "StateType": "open",
      "EscalationResponseTime": 0,
      "UnlockTimeout": 0,
      "EscalationSolutionTime": 0,
      "LockID": 1,
      "TicketNumber": "1001936",
      "ArchiveFlag": "n",
      "Lock": "unlock",
      "CreateTimeUnix": 1404173292,
      "SLAID": "",
      "CustomerUserID": "customer"
    }
  ]
}
```

11.7.2.4.3. Ticket aktualisieren

cURL Kommando:

```
curl "http://localhost/otrs/nph-genericinterface.pl/Webservice/GenericTicketConnectorREST/Ticket/1686?UserLogin=agent&Password=123" -H "Content-Type: application/json" -d '{"Ticket\":"Queues\":"Postmaster\"}' -X PATCH
```

Antwort:

```
{
  "TicketNumber": "1001936",
  "TicketID": "1686"
}
```

11.7.2.4.4. Suche nach Tickets

cURL Kommando:

```
curl "http://localhost/otrs/nph-genericinterface.pl/Webservice/GenericTicketConnectorREST/Ticket?UserLogin=agent&Password=123&Queue=Postmaster"
```

Antwort:

```
{
  "TicketID": [
    "1686",
    "102",
    "100",
    "1"
  ]
}
```

12. The OTRS Daemon

The OTRS Daemon is an independent set of system processes that plan and execute tasks in background, either on a recurrent basis or triggered by events. OTRS Daemon is fundamental for the correct system operation.

In previous versions of OTRS (from 3.1 to 4) there was another process called OTRS Scheduler that does part of the work that the OTRS Demon do in OTRS 5. This old process is replaced by the OTRS Daemon which was re-written from the ground to make it more stable, scalable and robust than its predecessor.

The OTRS Daemon is capable to handle up to 10 tasks at the same time and it can work cooperatively with other OTRS Daemons on different frontend servers in a cluster environment.

When idle OTRS Daemon consist in five processes:

- The main daemon (`bin/otrs.Daemon.pl`)

This process is in charge to start and keep running the other children daemons.

- Task worker daemon (Kernel/System/Daemon/DaemonModules/SchedulerTaskWorker.pm)

This daemon executes all tasks that have in a list, in a first in first out basis. It can handle simultaneous tasks by creating its own children processes and it checks the task list several times per second. The task list can be filled by task manager daemons, event handlers, and other parts of the system.

Its main mission is to handle all the tasks in the list as soon as possible.

- Future task manager daemon (Kernel/System/Daemon/DaemonModules/SchedulerFutureTaskManager.pm)

This daemon checks for non recurring tasks that are set to be executed in the future (e.g. when a Generic Interface invoker tries to reach a server and it can't, a task could be set to schedule for execution in the next 5 minutes). At the correct time it sends it the task worker daemon.

- Cron task manager daemon (Kernel/System/Daemon/DaemonModules/SchedulerCronTaskManager.pm)

This daemon calculates when is the next execution time of all recurring tasks (e.g. a cache cleanup one time per week). This kind of tasks are specified in the SysConfig. At the right time for each task it sends the required information to the task worker daemon to execute them.

Anmerkung

If a task execution time definition is changed in SysConfig, it might take up to an hour for the daemon to pick up the change automatically. Alternatively the OTRS Daemon can be restarted to apply the change immediately.

- Generic Agent task manager daemon (Kernel/System/Daemon/DaemonModules/SchedulerGenericAgentTaskManager.pm)

This daemon scans for Generic Agent jobs stored in the database that have a time schedule (discarding all other Generic Agent jobs that are set to executed by events). When is time to run a Generic Agent job it sends the task information to the task worker daemon to handle the task.

Anmerkung

The number of active processes depends on the number of tasks that the OTRS Daemon is executing simultaneously in a time frame.

By default the each daemon logs all error messages on a separated file located in /opt/otrs/var/log/Daemon/*.log. These logs are kept in the system for a defined period. To change this behavior and/or to also log the non error messages, please update SysConfig settings in Daemon -> Core: :Log.

When a task could not be executed successfully for any reason, an email is sent to a predefined recipient reporting the issue. The content of the email includes the error messages and trace (if available).

The OTRS Daemon is an automated process that normally does not require human interaction. However it is possible to query its status and start or stop it if needed.

To be sure that the OTRS Daemon is running there is a Cron job that constantly checks that the process is alive. The main daemon is prepared to work even without a database connection, so is perfectly safe if the Cron task to start it is executed even before the database process in the system startup, and it is also tolerant to database disconnections.

If for any reason the OTRS Daemon needs to be stop (for example during a system maintenance), all unhandled tasks are saved, and as soon as the process is started again it continues with all pending tasks. For recurring tasks it will only execute the last instance of the task (if its due time was during the downtime).

12.1. OTRS Daemon Graphical Interface

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

When the system detects that the OTRS Daemon is not running, a notification is presented to a defined group of users ("admin" by default).

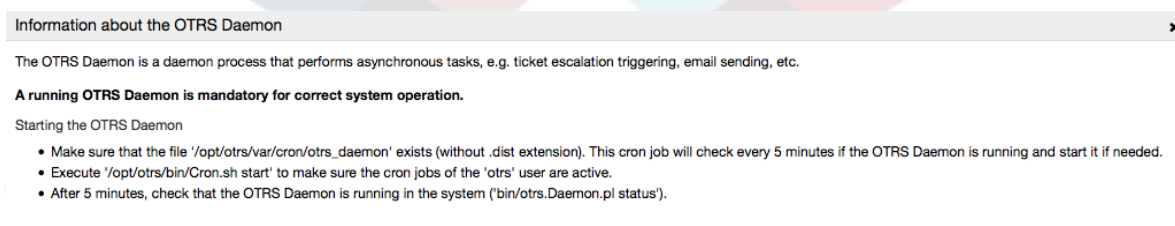
To disable the notification (not recommended), change or add the notification groups, please edit the Frontend: :NotifyModule###800-Daemon-Check setting in the SysConfig.

Abbildung 4.114. Daemon notification



Clicking the notification the system presents an overlay window explaining the steps to bring the OTRS Daemon up and running.

Abbildung 4.115. Start Daemon



12.2. OTRS Daemon Command Line Interface

The OTRS Daemon command line tools let you control the main daemon process (Start / Stop) or query its status. There are also tools to get more detailed information about the other four children daemons.

12.2.1. Main Daemon Tools

To start, stop or query daemon status bin/otrs.Daemon.pl script is used.

Beispiel 4.30. Example to start the OTRS Daemon

```
shell> cd /opt/otrs/
shell> OTRS_HOME/bin/otrs.Daemon.pl start
```

Verfügbare Optionen

- **start** - to start the OTRS Daemon process.
- **stop** - to stop the OTRS Daemon process.

- **status** - to query the OTRS Daemon process status.
- **start --debug** - to start the OTRS Daemon process in debug mode.

In this mode each daemon reports different messages depending on the actions that are been executed. This mode is not recommended for production environments.

- **stop --force** - to stop the OTRS Daemon process in reducing the wait for children processes to finish.

A forced stop reduces the amount of time the main daemon waits to successful stop the other children processes from 30 seconds (normal) to 5 seconds (forced).

12.2.2. Other Daemon Tools

To list all configured child daemons that the main daemon should start and keep running use the console command: `Maint::Daemon::List`.

Beispiel 4.31. Example to list all configured daemons

```
shell> cd /opt/otrs/  
shell> bin/otrs.Console.pl Maint::Daemon::List
```

To list detailed information of all daemons use the console command: `Maint::Daemon::Summary`.

Beispiel 4.32. Example to a summary of all daemon tasks

```
shell> cd /opt/otrs/  
shell> bin/otrs.Console.pl Maint::Daemon::Summary
```

Kapitel 5. Anpassung

1. Access Control Lists (ACLs)

1.1. Einführung

Seit OTRS 2.0 können Access Control Lists (ACLs) verwendet werden, um den Zugriff auf Tickets, Module, Queues, usw. zu steuern bzw. um in bestimmten Situationen Aktionen auf Tickets (schließen, verschieben, usw.) zu beeinflussen. ACLs können als Ergänzung zum bestehenden Berechtigungssystem der Rollen und Gruppen verwendet werden. Mit ACLs lassen sich, basierend auf Ticket-Attributen, rudimentäre Workflows innerhalb des Systems abbilden.

In a general way ACLs are used to reduce the possible options for a ticket based on a defined set of rules.

ACLs can be directly entered into the Kernel/Config.pm file. However this is not any more recommended as OTRS comes now with a GUI Access Control Lists in the Admin panel that allows to save the ACLs in the Database as the first step and then deploy them into a file when they are ready.

This chapter has some ACL examples which will walk you through the process of defining ACL definitions, and a reference of all possible important ACL settings.

Warnung

The default user 'root@localhost' is not affected by the Ticket ACLs

1.2. Definition

The ACL definition can be split into two big parts, 'Matching' and 'Change'. In the matching sections the ACLs contains attributes that has to be met in order to use the ACL. If the attributes defined in the ACL does not match with the attributes that are sent, then the ACL does not take any affect, but any other match ACL will. The change sections contains the rules to reduce the possible options for a ticket.

Filtersektionen

- Eigenschaften

This section contains matching options that can be changed on the fly. For example on a ticket creation time the data of the ticket changes dynamically as the agent sets the information. If an ACL is set to match a ticket attribute then only when the matching attribute is selected the ACL will be active and might reduce other ticket attributes, but as soon as another value is selected the ACL will not take any affect.

- PropertiesDatabase

This section is similar to 'Properties' but does not take changes in ticket attributes that are not saved into the DataBase, this means that changing an attribute without submit will not make any effect. This section is not use for ticket creation screens (as tickets are not yet created in the Database).

Änderungssektionen

- Possible

Possible section resets the data to be reduced to only the elements that are set in this section.

- PossibleAdd

Elements in PossibleAdd section add missing elements that were reduced in other ACLs. PossibleAdd is only used in together with other ACLs that have Possible or PossibleNot sections.

- PossibleNot

This section is used to remove specific elements from the current data. It could be used stand alone or together with other ACLs with a Possible or PossibleAdd sections.

In order to make the development of ACLs easier and more powerful there is a set of so called modifiers for the attributes on each section. This modifiers are explained below:

Modifikator

- [Not]

This modifier is used to negate a value for example: '[Not]2 low' in this case talking about ticket priorities will be the same as to have: '1 very low', '3 normal', '4 high', '5 very high'.

- [RegExp]

It is use to define a regular expression for matching several values, for example '[RegExp]low' talking about priorities is the same as '1 very low', '2 low'.

- [regex]

Es ist sehr ähnlich zu [RegExp], aber es unterscheidet zwischen Groß-/Kleinschreibung.

- [NotRegExp]

Negated regular expressions for example '[NotRegExp]low' talking about priorities is the same as '3 normal', '4 high', '5 very high'.

- [Notregex]

Es ist ziemlich ähnlich zu [NotRegExp], aber es unterscheidet nicht zwischen Groß-/Kleinschreibung

1.3. Beispiele

Die folgenden Beispiele sind für beide Arten, grafisch und textbasiert.

Beispiel 5.1. ACL, die das Verschieben von Tickets in eine Queue nur bei einer Ticket-Priorität von 5 erlaubt.

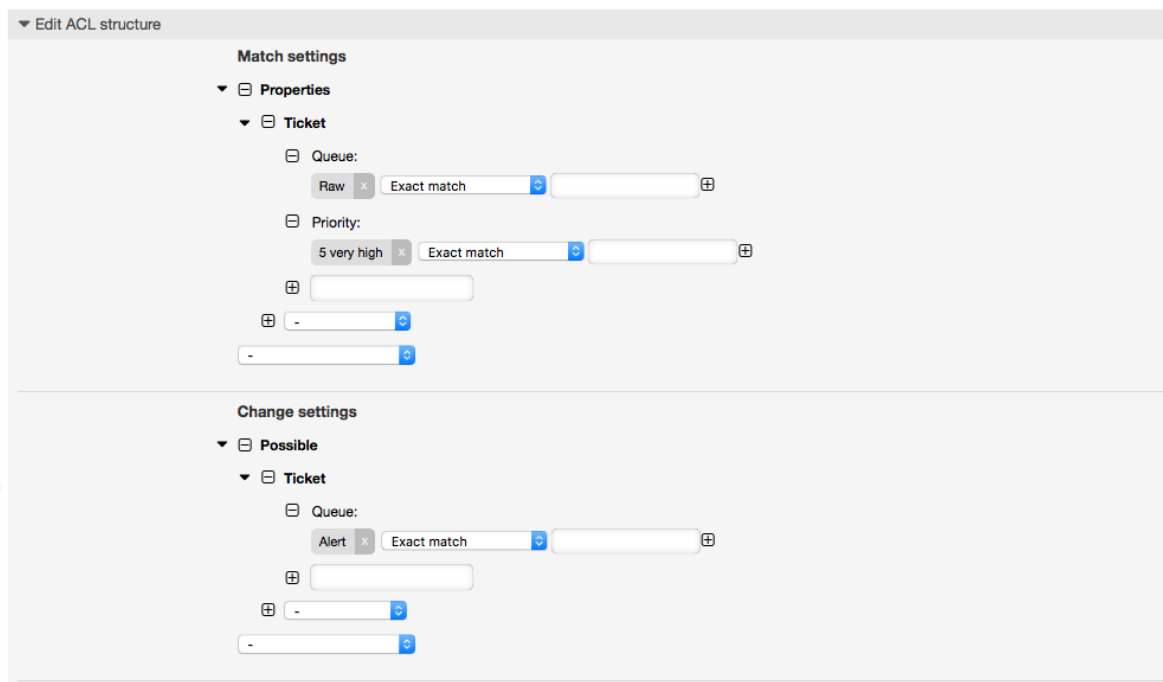
This example shows you the basic structure of an ACL. First, it needs to have a name. In this case, it is "100-Example-ACL". Note that the ACLs will be numerically sorted before execution, so you should use the names carefully.

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" at this moment the ACL will not match, but then priority drop-down is selected and the priority is changed now to "5 very high" then 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.

Abbildung 5.1. ACL 100-Example-ACL



```
# 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 5.2. ACL, die das Verschieben von Tickets in eine Queue nur bei einer Ticket-Priorität von 5 erlaubt.

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.

Abbildung 5.2. ACL 102-Example-ACL

▼ Edit ACL structure

Match settings

- ▼ PropertiesDatabase
 - ▼ Ticket
 - Queue:
 - Raw Exact match
 - Priority:
 - 5 very high Exact match
 -
 -

Change settings

- ▼ Possible
 - ▼ Ticket
 - Queue:
 - Alert Exact match
 -
 -

```
# ticket acl
$self->{TicketAcl}->{'102-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'],
    },
  },
};
```

Beispiel 5.3. 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.

Abbildung 5.3. ACL 102-Second-Example-ACL

▼ Edit ACL structure

Match settings

- ▼ Properties
 - ▼ Ticket
 - Queue:
 - Raw Exact match

Change settings

- ▼ Possible
 - ▼ Ticket
 - State:
 - new open pending reminder Exact match
- ▼ PossibleNot
 - ▼ Action
 - AgentTicketClose Exact match

```

$Self->{TicketAcl}->{'102-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'],
    },
  },
  # return also not possible options (black list)
  PossibleNot => {
    # not possible action options
    Action => [ 'AgentTicketClose' ],
  },
};

```

Beispiel 5.4. ACL entfernt immer den Status Erfolgreich geschlossen

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).

Abbildung 5.4. ACL 103-Third-ACL-Example

▼ Edit ACL structure

Match settings

-

Change settings

- ▼ PossibleNot
 - ▼ Ticket
 - State:
 - closed successful x Exact match
 -
 -
 -

```
$Self->{TicketAcl}->{'103-Third-ACL-Example'} = {
  # match properties
  Properties => {
    # current ticket match properties (match always)
  },
  # return possible options
  PossibleNot => {
    # possible ticket options
    Ticket => {
      State => ['closed successful'],
    },
  },
};
```

Beispiel 5.5. ACL, die nur Hardware-Services für Tickets anzeigt, die in Queues erstellt wurden, die mit "HW" beginnen.

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

Abbildung 5.5. ACL 104-Only-Hardware-Services-for-HW-Queues-ACL

▼ Edit ACL structure

Match settings

- ▼ Properties
 - ▼ Ticket
 - Queue:
 - [RegExp]HW x Regex
 -
 -
 -

Change settings

- ▼ Possible
 - ▼ Ticket
 - Queue:
 - [RegExp]^Hardware x Regex
 -
 -
 -

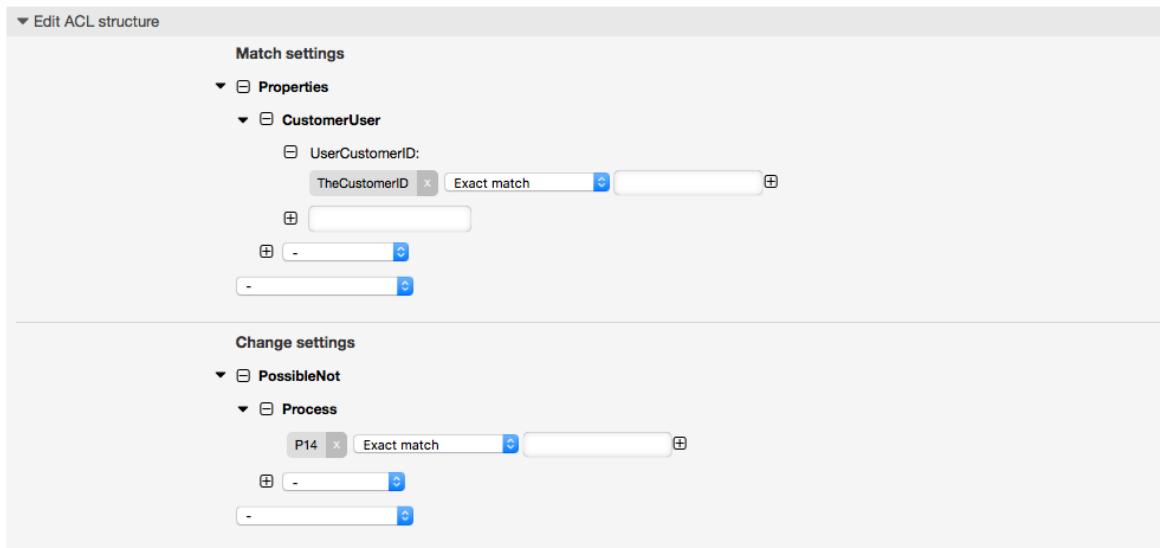
```

$Self->{TicketAcl}->{'104-Only-Hardware-Services-for-HW-Queues-ACL'} = {
  # 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)'],
    },
  },
};

```

Beispiel 5.6. ACL to restrict a Process in the customer frontend using the CustomerID.

Abbildung 5.6. ACL 105-Disallow-Process-For-CustomerID



```

$Self->{TicketAcl}->{"105-Disallow-Process-For-CustomerID"} = {
  'Possible' => {},
  'PossibleNot' => {
    'Process' => [
      'P14'
    ]
  },
  'Properties' => {
    'CustomerUser' => {
      'UserCustomerID' => [
        'CustomerID'
      ]
    }
  },
  'PropertiesDatabase' => {},
  'StopAfterMatch' => 0
};

```

1.4. Referenz

In dem angehängten Beispiel finden Sie eine Liste aller Parameter, welche für ACLs verwendet werden können.

Please see the section on ACLs in the ProcessManagement documentation for a detailed description of how to use ACLs for process tickets.

Beispiel 5.7. Die Liste zeigt alle möglichen und wichtigen ACL-Einstellungen

```
# 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'],
      UserCustomerID => ['some customer id'],
      Group_rw => [
        'hotline',
      ],
      Role => [
        'admin',
      ],
      # ...
    },

    # process properties
    Process => {
      ProcessEntityID => ['Process-9c378d7cc59f0fce4cee7bb9995ee3eb'],
      # the Process that the current ticket is part of
      ActivityEntityID => ['Activity-f8b2fdebe54eeb7b147a5f8e1da5e35c'],
      # the current Activity of the ticket
      ActivityDialogEntityID => ['ActivityDialog-aff0ae05fe6803f38de8fff6cf33b7ce'],
      # the current ActivityDialog that the Agent/Customer is using
    },

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

```

    # ...
  },
  Type => {
    ID => ['some id'],
    Name => ['some name'],
    # ...
  },
  Priority = {
    ID => ['some id'],
    Name => ['some name'],
    # ...
  },
  SLA = {
    SLAID => ['some id'],
    Name => ['some name'],
    Calendar => ['some calendar'],
    # ...
  },
  State = {
    ID => ['some id'],
    Name => ['some name'],
    TypeName => ['some state type name'],
    TypeID => ['some state type id'],
    # ...
  },
  Owner => {
    UserLogin => ['some login'],
    Group_rw => [
      'some group',
    ],
    Role => [
      'admin',
    ],
    # ...
  },
  Responsible => {
    UserLogin => ['some login'],
    Group_rw => [
      'some group',
    ],
    Role => [
      'admin',
    ],
    # ...
  },
  DynamicField => {
    # Names must be in DynamicField_<field_name> format.
    # Values in [ ... ] must always be the untranslated internal data keys
    # specified in the dynamic field definition and
    # not the data values shown to the user.
    DynamicField_Field1 => ['some value'],
    DynamicField_OtherField => ['some value'],
    DynamicField_TicketFreeText2 => ['some value'],
    # ...
  },
  # alternatively, ticket properties can be specified in the ticket hash
  Ticket => {
    Queue => ['Raw'],
    State => ['new', 'open'],
    Priority => ['some priority'],
    Lock => ['lock'],
    CustomerID => ['some id'],
    CustomerUserID => ['some id'],
    Owner => ['some owner'],
    DynamicField_Field1 => ['some value'],
    DynamicField_MyField => ['some value'],
    # ...
  },
},
# match properties (existing values from the database)
PropertiesDatabase => {

```



```

    # See section "Properties", the same config can be used here.
    # ...
  }

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

    # Limit the number of possible ActivityDialogs the Agent/Customer
    # can use in a process ticket.
    ActivityDialog => ['AD1', 'AD3'],

    # Limit the number of possible Processes that can be started
    Process => ['Process-9c378d7cc59f0fce4cee7bb9995ee3eb',
'Process-12345678901234567890123456789012'],

    # possible action options (white list)
    Action => [
      'AgentTicketBounce',
      'AgentTicketPhone'.      # only used to show/hide the Split action
      'AgentLinkObject',      # only used to show/hide the Link action
      # ...
    ],
  },
  # add options (white list)
  PossibleAdd => {
    # See section "Possible"
    # ...
  },
  # 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.

From OTRS 4 the 'Action' parameter is not longer a hash but an array reference and it can be used in any of the Change sections using any of the Modifiers.

2. Prozess-Management

2.1. Einführung

Diese Funktion erlaubt Ihnen in OTRS Prozesse (Work-Flows) zu modellieren. Die grundlegende Idee ist wiederkehrende Prozesse zu definieren und die Arbeit an verschiedene Personen zu delegieren, genauso um den Prozess in verschiedene Richtungen, basierend auf bestimmten Kriterien, zu führen.

2.2. Beispiel Prozess

Zur Demonstration schauen wir uns ein Beispiel an. Wir wollen die Bestellung eines Buches in einem Prozess abbilden.

2.2.1. Aufnahme der Anforderungen

Bevor eine Bestellung abgesetzt wird, wird die Anforderung eines Mitarbeiters von Literatur gespeichert. Das folgende Buch wird in unserem Beispiel benötigt:

Titel: Prozessmanagement für Dummies Autor: Thilo Knuppertz ISBN: 3527703713
--

2.2.2. Genehmigung durch Manager

Der Abteilungsleiter des Mitarbeiters muss über die Bestellung entscheiden. Im Falle einer Verweigerung sollte ein Grund vom Manager angegeben werden. Im Falle einer Genehmigung wird die Bestellung zur Einkauf-Abteilung weitergeleitet.

2.2.3. Verarbeitung durch Einkaufsabteilung

Die Einkauf-Abteilung muss nun herausfinden, wo das Buch zu den besten Konditionen bestellt werden kann. Wenn es ausverkauft ist kann das in der Bestellung vermerkt werden. Im Falle einer erfolgreichen Bestellung wird die Einkauf-Abteilung den Lieferanten, den Preis und das Lieferdatum aufnehmen.

2.2.4. Verarbeitung durch Poststelle

Die Lieferung kommt an der Firma an. Die Warenannahme überprüft die Lieferung und notiert das Empfangsdatum. Nun wird der Mitarbeiter informiert, dass seine Bestellung angekommen ist und bereit ist, abgeholt zu werden.

2.3. Implementierung des Beispiels

Wenn wir davon ausgehen, dass sich ein Ticket in diesem Work-Flow wie ein Begleitdokument verhält, das Veränderungen als Notizen empfängt, dann haben wir schon ein klares Bild der Prozess-Tickets.

Nach der Analyse des Beispielprozesses, konnten wir folgende nötigen Elemente identifizieren:

- Möglichkeiten zur Datenaufnahme, hier *Aktivitäten-Dialoge*,
- Kontrollen, die automatisch auf geänderte Daten reagieren kann, nennen wir sie *Übergänge*,

- Änderungen, die sich auf ein Prozess-Ticket nach erfolgreichem Übergang eines Prozess-Ticket angewendet werden können, nennen wir sie *Übergangsaktionen*.

Wir brauchen noch ein zusätzliches Element, was nicht so offensichtlich sein könnte:

- Eine Möglichkeit mehr als nur einen Aktivitätsdialog zur Verfügung zu stellen. In unserem Beispiel ist das erforderlich, wenn die Führungskraft zwischen "Genehmigen" und "Ablehnen" zu entscheiden hat. Nennen wir sie *Aktivität*.

Mit Aktivitäten, Aktivitätsdialogen, Übergängen und Übergangsaktionen haben wir die notwendigen Werkzeuge, um die einzelnen Schritte unseres Beispiels zu modellieren. Was noch fehlt, ist ein Bereich, wo für jeden Arbeitsablauf die Reihenfolge der Schritte festgelegt wird. Nennen wir diesen *Prozess*. Um uns auf all diese Personen später beziehen zu können, werden wir ihnen in Klammern eine Abkürzung zuweisen. Diese Abkürzungen basieren auf einem internen Identifikations-Mechanismus namens EntityIDs.

Die EntityIDs werden mit einem oder zwei Buchstaben (je nach Prozessteil) und einer fortlaufenden Nummer gebildet, Beispiel:

- Prozess: 'P1', 'P2' ... 'Pn'.
- Aktivität: 'A1', 'A2' ... 'An'.
- Aktivitätsdialog: 'AD1', 'AD2' ... 'ADn'.
- Übergang: 'T1', 'T2' ... 'Tn'.
- Übergangsaktivität: 'TA1', 'TA2' ... 'TAn'.

Vor der Erstellung des Prozesses und seiner Abschnitte ist es notwendig, das System vorzubereiten, wir müssen sowohl einige Queues, Nutzer und Dynamic Fields definieren, als auch ein paar SysConfig-Optionen festlegen.

Erstellen Sie die folgenden Queues:

- Management
- Mitarbeiter
- Einkauf
- Post

Erstellen Sie die folgenden Benutzer:

- Manager
- Mitarbeiter

Erstellen Sie die folgenden DynamicFields

- Titel

Beschriftung	Titel
Typ	Text
Objekt	Ticket

- Autor

Beschriftung	Autor
Typ	Text
Objekt	Ticket

- ISBN

Beschriftung	ISBN
Typ	Text
Objekt	Ticket

- Status

Beschriftung	Status
Typ	Einfachauswahl
Objekt	Ticket
mögliche Werte	<ul style="list-style-type: none"> • Genehmigung • Genehmigung abgelehnt • genehmigt • Bestellung abgelehnt • Bestellung aufgegeben • Sendung empfangen

Bemerkung: Bitte benutzen Sie exakt diese möglichen Werte für "Schlüssel" und "Wert" in den DynamicField-Einstellungen

- Lieferant

Beschriftung	Lieferant
Typ	Text
Objekt	Ticket

- Preis

Beschriftung	Preis
Typ	Text
Objekt	Ticket

- LieferDatum

Beschriftung	Liefertermin
Typ	Datum
Objekt	Ticket

- Eingangsdatum

Beschriftung	Datum des Eingangs
Typ	Datum

Objekt	Ticket
--------	--------

Setzen Sie die folgenden SysConfig-Einstellungen:

- 'Ticket::Responsible': Ja
- 'Ticket::Frontend::AgentTicketZoom###ProcessWidgetDynamicFieldGroups':

Schlüssel:	Inhalt:
Buch	Titel, Autor,ISBN
Allgemein	Status
Auftrag	Preis, Lieferant,Liefertermin
Lieferung	Eingangsdatum

- 'Ticket::Frontend::AgentTicketZoom###ProcessWidgetDynamicField':

Schlüssel:	Inhalt:
Autor	1
Eingangsdatum	1
LieferDatum	1
ISBN	1
Preis	1
Status	1
Lieferant	1
Titel	1

Fangen wir mit dem realen Prozessmanagement an. Im nächsten Schritt werden wir die einzelnen Objekte, die wir benötigen, definieren.


2.3.1. Prozess (als ein Container)

Um einen neuen Prozess zu erstellen ist es notwendig auf den Link "Prozess Management" in der Systemverwaltungs-Box im Admin-Menü zu klicken, anschließend gelangen wir zur Prozessmanagement-Übersicht. Nachdem der Prozess erstellt wurden, können wir alle anderen Objekte (oder Prozessteile) erstellen.

Anmerkung

Aktivitäten, Aktivitätsdialoge, Übergänge und Übergangsaktionen die in einem Prozess definiert wurden, sind ebenfalls für alle anderen Prozesse im System verfügbar.

Abbildung 5.7. OTRS Admin screen - System Administration

	System Administration	Online Admin Manual 
et types.	GenericAgent Manage tasks triggered by event or time based execution.	System Registration Manage system registration.
et states.	OTRS Business Solution™ Deploy and manage OTRS Business Solution™.	Support Data Collector Manage support data.
ices.	Cloud Services Manage OTRS Group cloud services.	Admin Notification Send notifications to users.
ts	Session Management Manage existing sessions.	System Maintenance Schedule a maintenance period.
vice Level	Performance Log View performance benchmark results.	System Log View system log messages.
	SQL Box Execute SQL statements.	Process Management Configure Processes.
	SysConfig Edit the system configuration settings.	Web Services Create and manage web services.
	Package Manager Update and extend your system with software packages.	Chat Channel Create and manage chat channels.

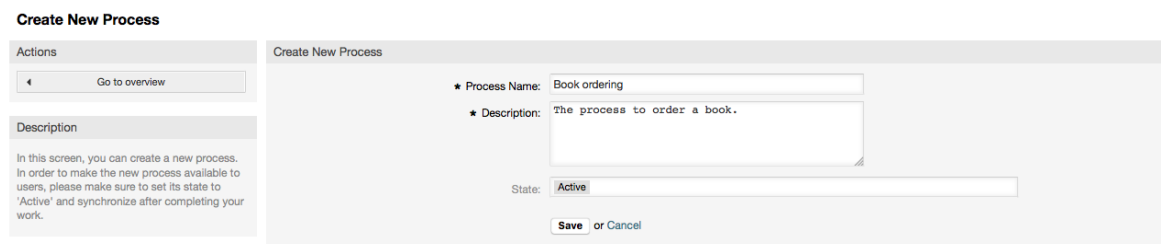
Klicken Sie auf "Neuen Prozess erstellen" im Adminmenü

Abbildung 5.8. Schaltfläche: Neuen Prozess erstellen



Füllen Sie die Prozessinformationen aus, definieren Sie einen Prozessname und die Beschreibung für diesen Prozess, wir belassen den Prozessstatus auf "inaktiv", bis wir alle Aufgaben erledigt haben. Speichern Sie den Vorgang.

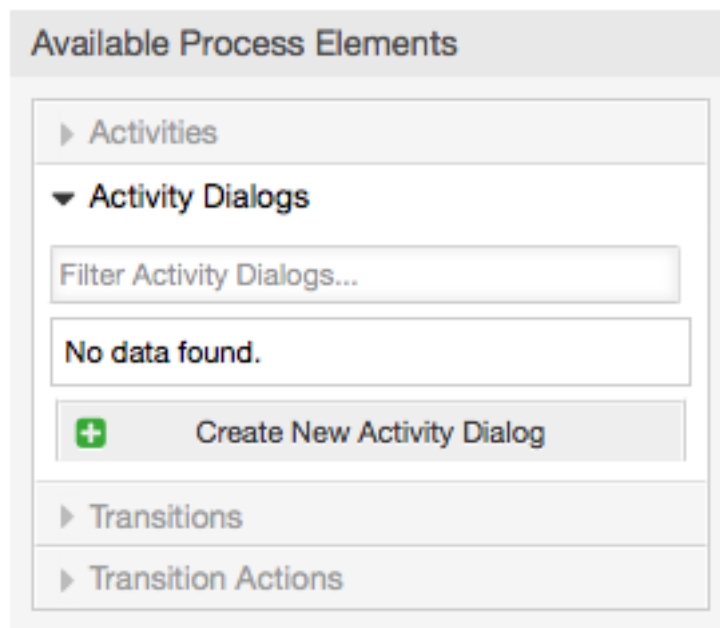
Abbildung 5.9. Neuen Prozess hinzufügen



2.3.2. Aktivitäts-Dialoge

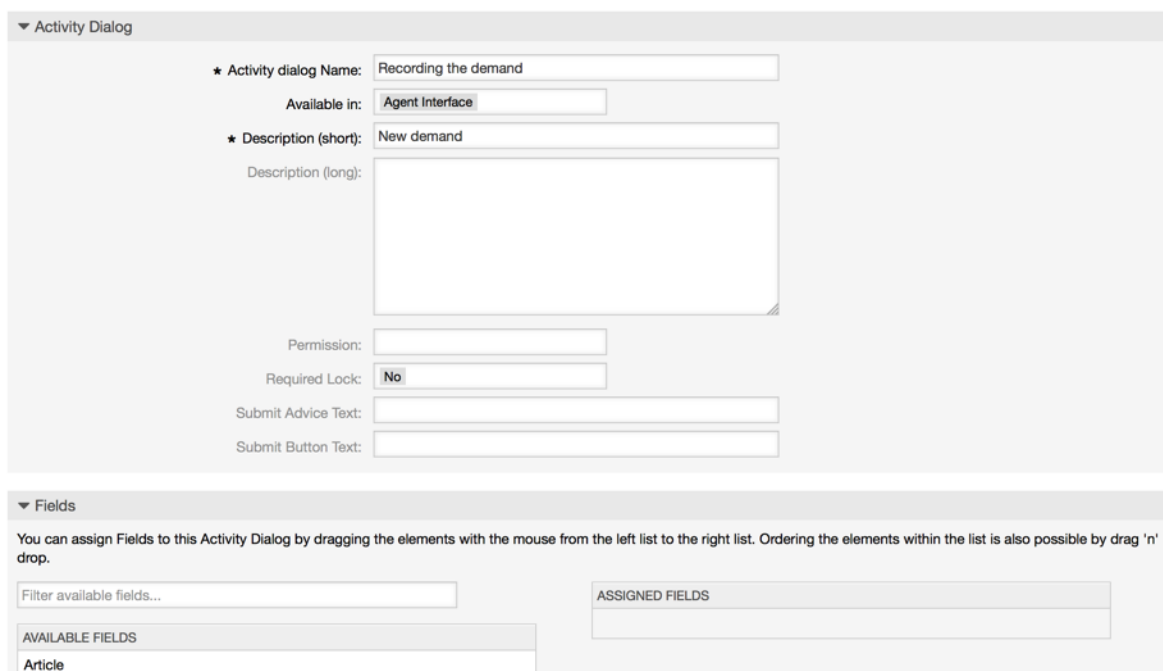
Click on the new process name in the Process Management Overview Screen, then in the "Available Process Elements" click in "Activity Dialogs" (this action will expand the activity dialog options and will collapse all others doing an accordion like effect), then click on "Create New Activity Dialog".

Abbildung 5.10. Neue Aktivitätsdialogschaltfläche erstellen



In the opened popup screen fill the "Activity dialog Name" as well as the "Description (short)" fields, for this example we will leave all other fields as the default, to assign fields to the Activity Dialog simply drag the required field from the "Available Fields" pool and drop into the "Assigned Fields" pool. The order in the "Assigned Fields" pool is the order as the fields will have in the screen, to modify the order simply drag and drop the field within the pool to rearrange it in the correct place.

Abbildung 5.11. Neuen Aktivitätsdialog hinzufügen



▼ Activity Dialog

★ Activity dialog Name:

Available in:

★ Description (short):

Description (long):

Permission:

Required Lock:

Submit Advice Text:

Submit Button Text:

▼ Fields

You can assign Fields to this Activity Dialog by dragging the elements with the mouse from the left list to the right list. Ordering the elements within the list is also possible by drag 'n' drop.

Filter available fields...

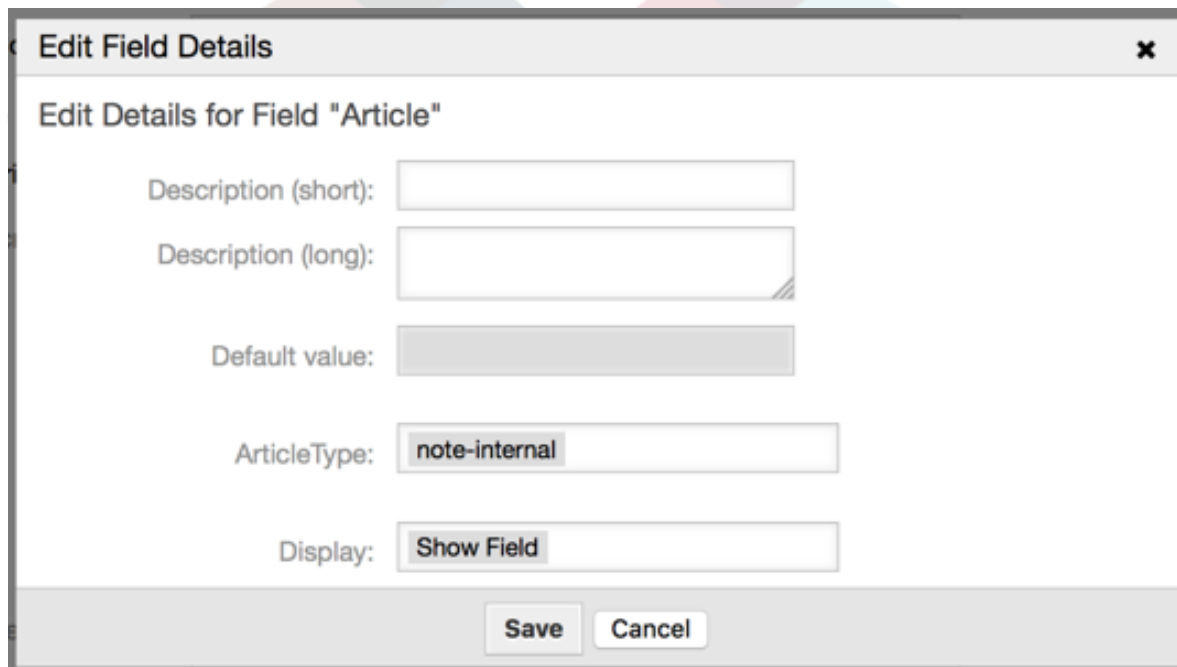
ASSIGNED FIELDS

AVAILABLE FIELDS

Article

As soon as the fields are dropped into the "Assigned Fields" pool another popup screen is shown with some details about the field, we will leave the default options and only for Article fields we should make sure that the ArticleType field is set to "note-internal".

Abbildung 5.12. Felddetails bearbeiten (Artikel)



Edit Field Details ✕

Edit Details for Field "Article"

Description (short):

Description (long):

Default value:

ArticleType:

Display:

Nachdem alle Felder zugewiesen wurden, klicken Sie auf übermitteln in der PopUp-Oberfläche damit die Änderungen gespeichert werden.

In this example we will use Article field for comments, but another option could be to create a TextArea type Dynamic Field, the rest of the mentioned fields in the lines below are the Dynamic Fields that we define before.

Please be aware that in this screen all the Dynamic Fields has the prefix "DynamicField_" as in "DynamicField_Title". Do not confuse with the field "Title" that is the Ticket Title.

Erstellen Sie die folgenden Aktivitätsdialoge:

- "Aufnahme der Nachfrage" (AD1)

Ein Aktivitätsdialog, welcher alle erforderlichen Felder für die Daten der Bestellung enthält (Titel, Autor und ISBN) und ein Statusfeld mit der Möglichkeit "Approval" zu wählen.

- "Genehmigung verweigert" (AD2)

Ein Aktivitätsdialog mit einem Kommentarfeld (Artikel) und einem Statusfeld mit der Möglichkeit "Approval denied".

- "Genehmigt" (AD3)

Hier benötigen Sie das Statusfeld mit der Option "Genehmigt".

- "Bestellung abgelehnt" (AD4)

Ein Aktivitätsdialog, welcher es ermöglicht eine unmögliche Bestellung abzulehnen (Buch ist nicht auf Lager). Hier wird ebenfalls ein Kommentarfeld und ein Statusfeld mit der Option "Order denied" benötigt.

- "Bestellung aufgegeben" (AD5)

Ein Aktivitätsdialog mit den Feldern Lieferant, Preis und Lieferdatum und dem Statusfeld mit der Option "Bestellung abgegeben".

- "Sendung empfangen" (AD6)

Eine Aktivität für die Poststelle mit einem Feld für das Empfangsdatum und dem Statusfeld mit der Option "Bestellung empfangen".

To restrict the Status field for each activity dialog we need to add some ACLs in the Kernel/Config.pm or to a new Perl file located in Kernel/Config/Files.

```

$self->{TicketAcl}->{'P1-AD1-1'} = {
    Properties => {
        Process => {
            ActivityDialogEntityID => ['AD1'],
        },
    },
    Possible => {
        Ticket => {
            DynamicField_Status => ['Approval'],
        },
    },
};

$self->{TicketAcl}->{'P1-AD2-1'} = {
    Properties => {
        Process => {
            ActivityDialogEntityID => ['AD2'],
        },
    },
    Possible => {
        Ticket => {
            DynamicField_Status => ['Approval denied'],
        },
    },
};

```

```

$Self->{TicketAcl}->{'P1-AD3-1'} = {
  Properties => {
    Process => {
      ActivityDialogEntityID => ['AD3'],
    },
  },
  Possible => {
    Ticket => {
      DynamicField_Status => ['Approved'],
    },
  },
};

$Self->{TicketAcl}->{'P1-AD4-1'} = {
  Properties => {
    Process => {
      ActivityDialogEntityID => ['AD4'],
    },
  },
  Possible => {
    Ticket => {
      DynamicField_Status => ['Order denied'],
    },
  },
};

$Self->{TicketAcl}->{'P1-AD5-1'} = {
  Properties => {
    Process => {
      ActivityDialogEntityID => ['AD5'],
    },
  },
  Possible => {
    Ticket => {
      DynamicField_Status => ['Order placed'],
    },
  },
};

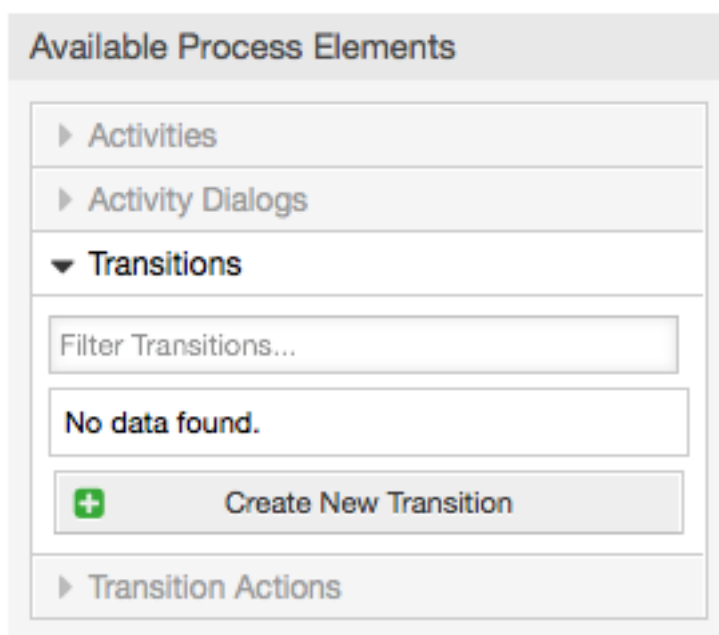
$Self->{TicketAcl}->{'P1-AD6-1'} = {
  Properties => {
    Process => {
      ActivityDialogEntityID => ['AD6'],
    },
  },
  Possible => {
    Ticket => {
      DynamicField_Status => ['Shipment received'],
    },
  },
};

```

2.3.3. Übergänge

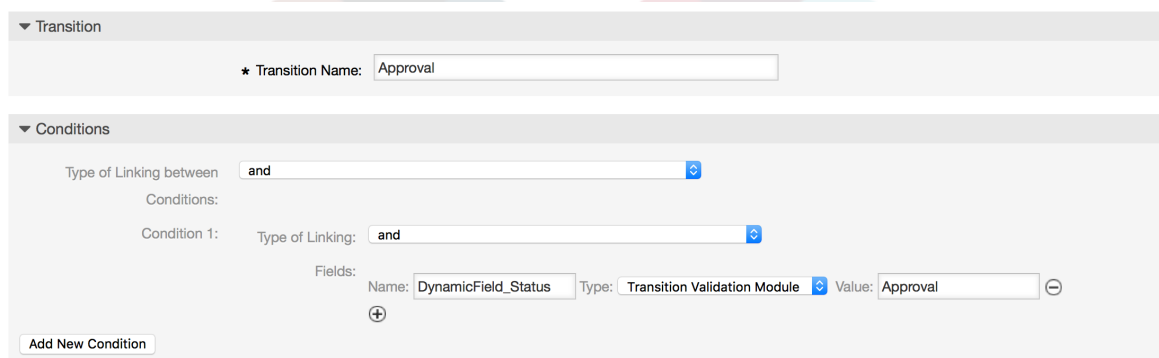
Klicken Sie in den "Verfügbaren Prozesselementen" auf "Übergänge", anschließend klicken Sie auf "Erstelle neue Übergans-Aktion".

Abbildung 5.13. Neuen Übergang hinzufügen



In dem öffnenden Popup füllen Sie bitte den "Übergangsnamen" aus, anschließend tragen Sie die benötigten Zustände ein. In diesem Beispiel werden wir nur einen Zustand und nur ein Feld benutzen. Für diese beiden lassen für den Linktyp auf "and" und wir werden den Match-Type-Wert als "String" nutzen.

Abbildung 5.14. Neuen Übergang hinzufügen



Nachdem alle Parameter und Werte gesetzt worden sind, klicken Sie auf die "Übermitteln"-Schaltfläche um die Werte zu speichern.

Erstellen Sie die folgenden Übergänge:

- "Genehmigung" (T1)

Ein Übergang, der überprüft ob das Statusfeld auf "Genehmigung" gesetzt ist.

- "Genehmigung verweigert" (T2)

Ein Übergang, der überprüft ob das Statusfeld auf "Genehmigung verweigert" gesetzt ist.

- "Genehmigt" (T3)

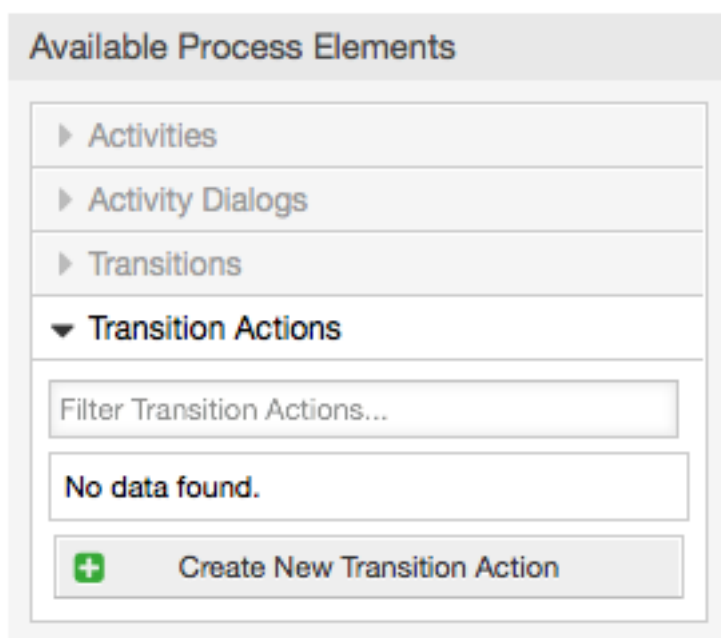
Ein Übergang, der überprüft ob das Statusfeld auf "Genehmigung erteilt" gesetzt ist.

- "Bestellung abgelehnt" (T4)
Ein Übergang, der überprüft ob das Statusfeld auf "Bestellung verweigert" gesetzt ist.
- "Bestellung aufgegeben" (T5)
Ein Übergang, der überprüft ob das Statusfeld auf "Bestellung aufgegeben" gesetzt ist.
- "Sendung empfangen" (T6)
Ein Übergang, der überprüft ob das Statusfeld auf "Sendung empfangen" gesetzt ist.

2.3.4. Übergangs-Aktionen

Klicken Sie auf "Übergangs-Aktionen" in den "Verfügbaren Prozesselementen", anschließend klicken Sie auf "Erstelle neue Übergangs-Aktion".

Abbildung 5.15. Erstellen Sie eine neue Schaltfläche "Übergangs-Aktion"



In the opened popup screen fill the "Transition Action Name", and the "Transition Action module" then add the required and optional parameter names and values.

All the Transition Action Modules are located in Kernel/System/ProcessManagement/TransitionAction and the following is the list of bundled Transition Actions included in this release:

- DynamicFieldSet
- TicketArticleCreate
- TicketCreate
- TicketCustomerSet
- TicketLockSet
- TicketOwnerSet
- TicketQueueSet

- TicketResponsibleSet
- TicketServiceSet
- TicketSLASet
- TicketStateSet
- TicketTitleSet
- TicketTypeSet

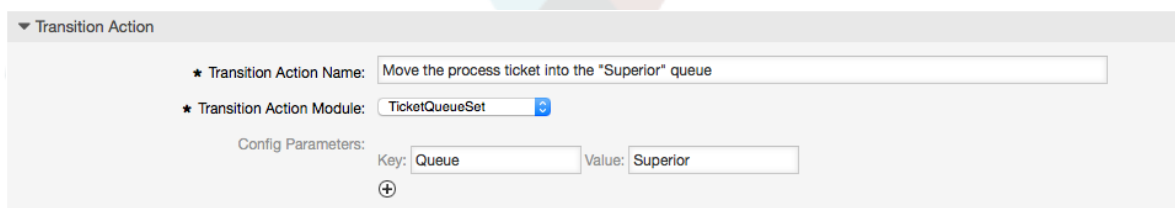
Jedes Modul hat seine eigenen und verschiedenen Parameter. Eine Übersicht aller benötigten und optionalen Parameter finden Sie unter Moduldokumentation.

Anmerkung

From OTRS 4.0.1, parameters are not longer mandatory fixed values, but instead, they can inherit from the original ticket using format: <OTRS_Ticket_property>.

From OTRS 4.0.6, the format <OTRS_TICKET_property> is now supported, older format is still usable, but deprecated as it will be dropped in further versions.

Abbildung 5.16. Neue Übergangs-Aktion hinzufügen



Nachdem alle Parameter und Werte gesetzt worden sind, klicken Sie auf die "Übermitteln"-Schaltfläche um die Werte zu speichern.

Erstellen Sie die folgenden Übergangs-Aktionen:

- "Verschiebe das Prozessticket in die Management-Queue" (TA1)
Diese Aktion wird vermutlich ausgeführt wenn der Übergang "Approval" (T1) angewandt wurde.
- "Ändere den Ticket-Verantwortlichen zu 'Manager'" (TA2)
Wird ausgeführt, wenn der Übergang "Approval" (T1) ausgeführt wurde.
- "Move process ticket into the 'Employees' queue" (TA3)
Wird ausgeführt, wenn:
 - Der Übergang "Approval denied" (T2) wird ausgeführt
 - Der Übergang "Order denied" (T4) wird ausgeführt.
 - Der Übergang "Shipment received" (T6) wird ausgeführt
- Ändere den Ticket-Verantwortlichen zu 'Employee'" (TA4)
Wird ausgeführt, wenn:
 - Der Übergang "Approval denied" (T2) wird ausgeführt
 - Der Übergang "Order denied" (T4) wird ausgeführt.

- Der Übergang "Shipment received" (T6) wird ausgeführt
- "Verschiebe das Prozessticket in die 'Purchasing'-Queue" (TA5)
Wird ausgeführt, wenn der Übergang "Approved" (T3) ausgeführt wurde.
- "Verschiebe das Prozessticket in die 'Post office'-Queue" (TA6)
Wird ausgeführt, wenn der Übergang "Order placed (T5) ausgeführt wurde.
- "Ticket erfolgreich schließen" (TA7)
Wird ausgeführt, wenn:
 - Der Übergang "Shipment received" (T6) wird ausgeführt
- Ticket erfolglos schließen" (TA8)
Wird ausgeführt, wenn:
 - Der Übergang "Approval denied" (T2) wird ausgeführt
 - Der Übergang "Order denied" (T4) wird ausgeführt.

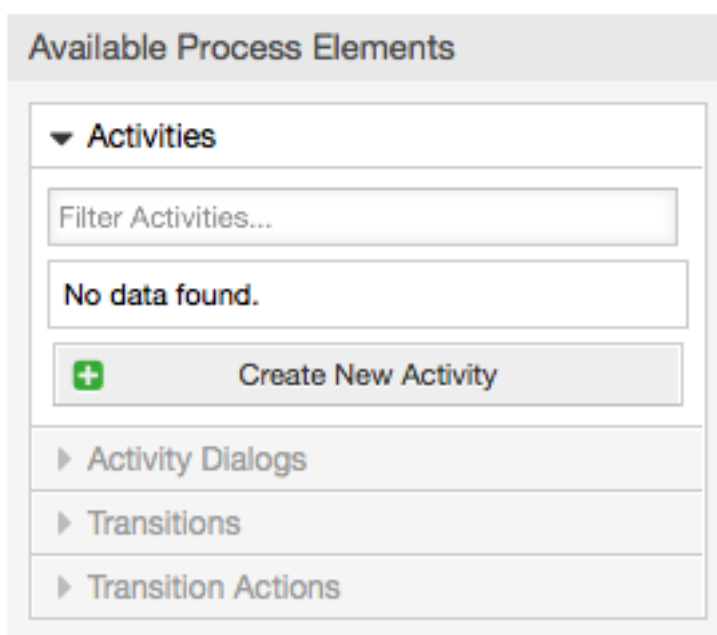
Wie Sie sehen, gibt es Orte an denen gleiche Übergangsaktionen ausgeführt werden sollen. Deshalb ist es sinnvoll Übergangsaktionen frei mit Übergängen zu verknüpfen, um sie wiederverwenden zu können.

2.3.5. Aktivitäten

We chose the approach to see Activities as a basket which can contain one or more Activity Dialogs.

Klicken Sie auf "Aktivitäten" in den verfügbaren "Prozesselementen", anschließend klicken Sie auf "Erstelle neue Aktivität".

Abbildung 5.17. Neue Aktivitätsschaltfläche erstellen



In the opened popup screen fill the "Activity Name", then drag the required Activity Dialogs from the "Available Activity Dialogs" pool, and drop them into to the "Assigned Activity Dialogs" pool. This dialogs will be presented (in the ticket zoom screen) in the same order as it is defined on this screen translating from top to bottom, from left to right.

This order is specially important in the first Activity, since the first Activity Dialog for this activity is the only one that is presented when the process starts.

Erstellen Sie die folgenden Aktivitäten:

- "Aufnahme der Nachfrage" (A1)
Beinhaltet die Aktivität "Aufnahme der Nachfrage" (AD1)
- "Genehmigung" (A2)
Beinhaltet die Aktivitätsdialoge "Genehmigung verweigert" (AD2) und "Genehmigt" (AD3)
- "Bestellung" (A3)
Beinhaltet die Aktivitätsdialoge "Bestellung abgelehnt" (AD4) und "Bestellung aufgegeben" (AD5)
- Eingehend (A4)
Beinhaltet den Aktivitätsdialog "Bestellung erhalten" (AD6)
- "Prozess beendet" (A5): Das ist eine Aktivität ohne mögliche Aktivitätsdialoge. Sie wird nach "Genehmigung verweigert", "Genehmigt", "Bestellung abgelehnt", "Bestellung aufgegeben" oder "Bestellung erhalten" angezeigt und stellt das Ende des Prozesses dar.

Now we can clearly see that Activities are precisely defined states of a process ticket. After a successful Transition a process ticket moves from one Activity to another.

2.3.6. Buchbestellprozesspfad

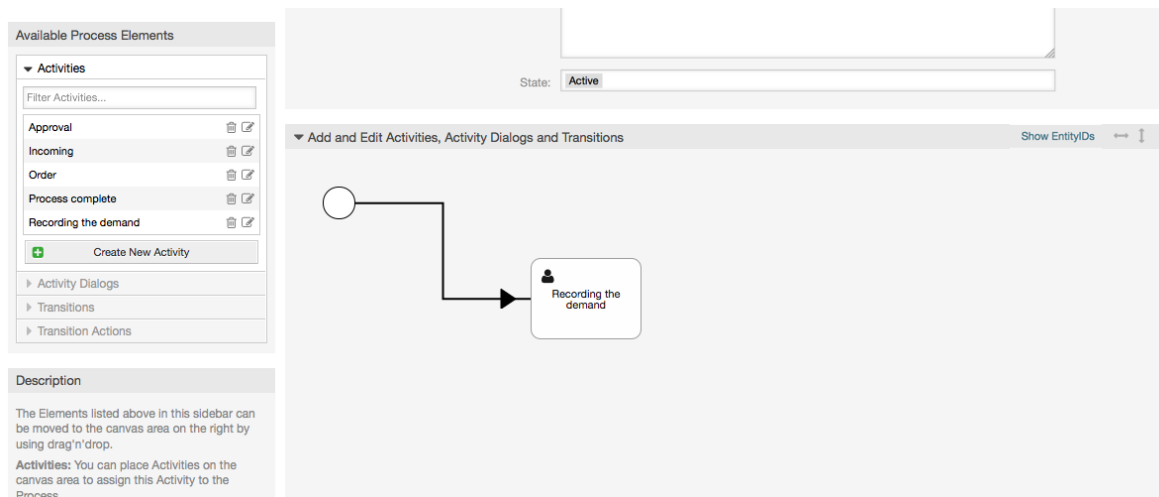
Let us conclude our example with the last missing piece in the puzzle, the Process as the a flow describer. In our case this is the whole ordering work-flow. Other processes could be office supply ordering or completely different processes.

The process has a starting point which consists of the start Activity and the start Activity Dialog. For any new book order, the start Activity Dialog (first Activity Dialog for the first Activity) is the first screen that is displayed. If this is completed and saved, the Process ticket will be created and can follow the configured work-flow.

The process also contains the directions for how the process ticket can move through the Process. Let's call this the "Path". It consists of the start Activity, one or more Transitions (possibly with Transition Actions), and other Activities.

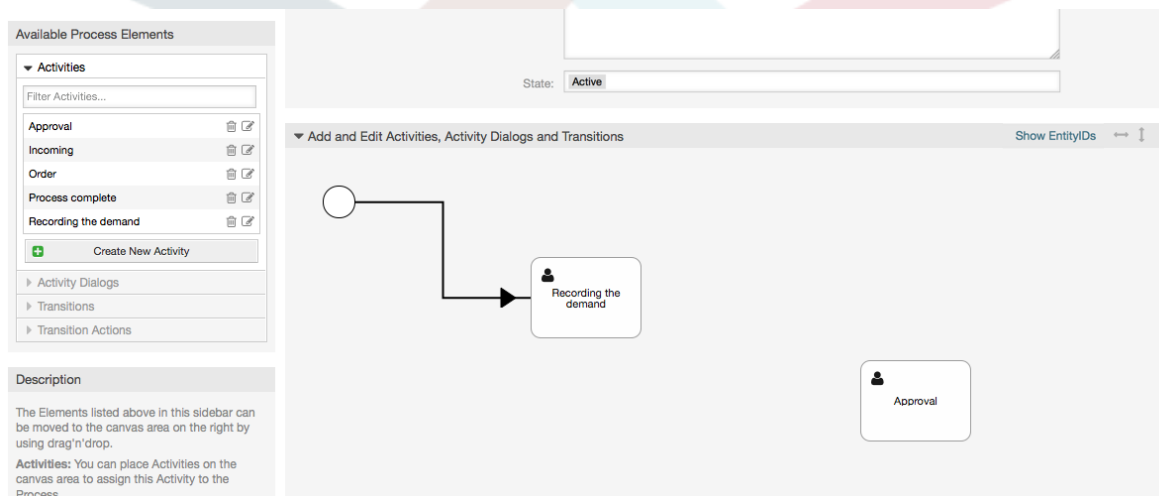
Assuming that the Activities has already assigned their Activity Dialogs drag an Activity from the accordion (in the left part of the screen) and drop it into the canvas area (below process information). Notice that an arrow from the process start (green circle) to the Activity is placed automatically. (This is the first Activity and its first Activity Dialog is the first screen that will be shown when the process starts).

Abbildung 5.18. Ziehen Sie die erste Aktivität in die Arbeitsfläche



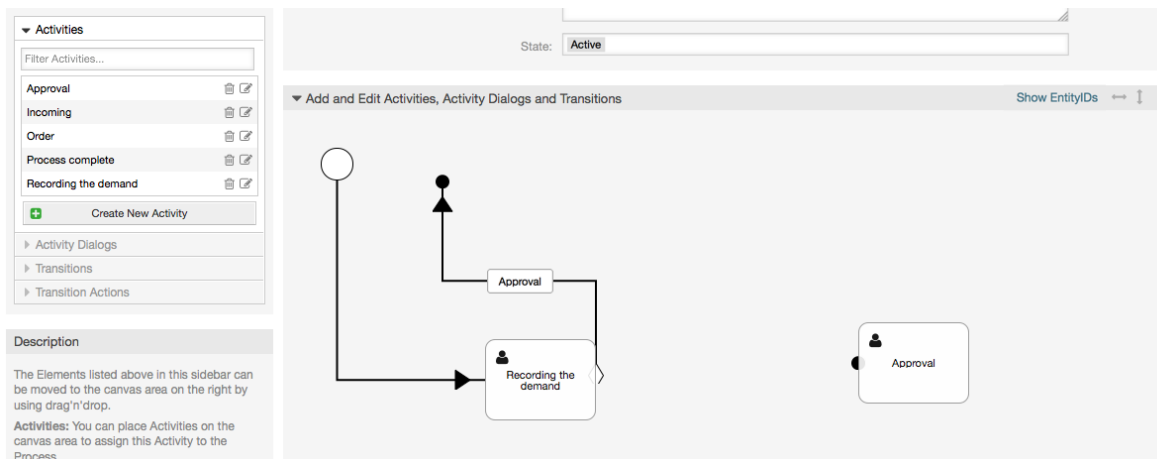
Next, drag another Activity into the canvas too. Now we will have two Activities in the canvas. The first one is connected to the start point and the second has no connections. You can hover the mouse over each activity to reveal their own Activity Dialogs.

Abbildung 5.19. Ziehen Sie die zweite Aktivität in die Arbeitsfläche.



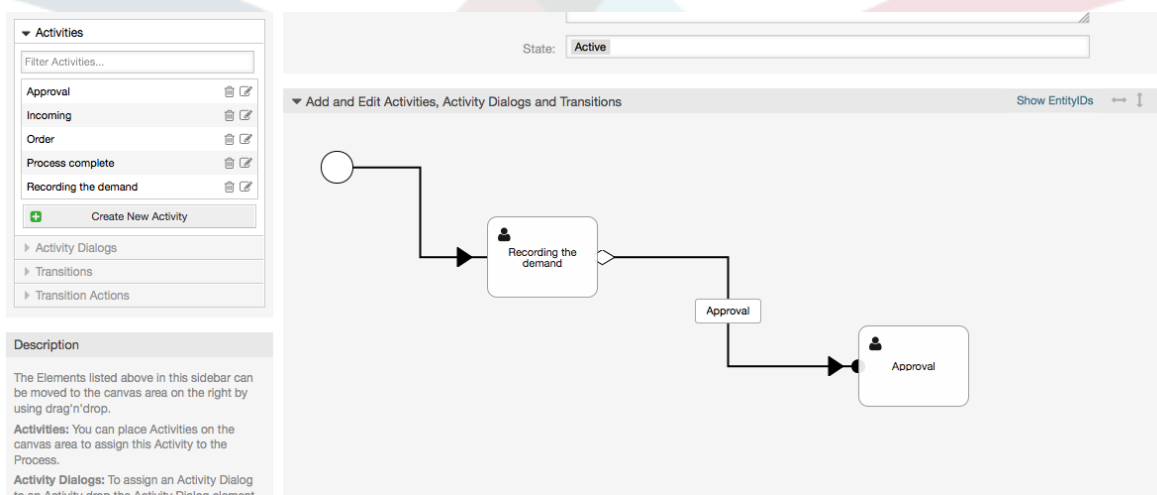
Then let's create the "Path" (connection) between this two Activities, for this we will use the Transitions. Click on Transitions in the accordion drag a Transition and drop it inside the first Activity. Notice that the Activity changes its color indicating that the Transition is attached. As soon as the Transition is dropped the end point of the Transition arrow will be placed next to the process start point. Drag the Transition arrow end point and drop it inside the other Activity to create the connection between the Activities.

Abbildung 5.20. Ziehen Sie einen Übergang in die Arbeitsfläche



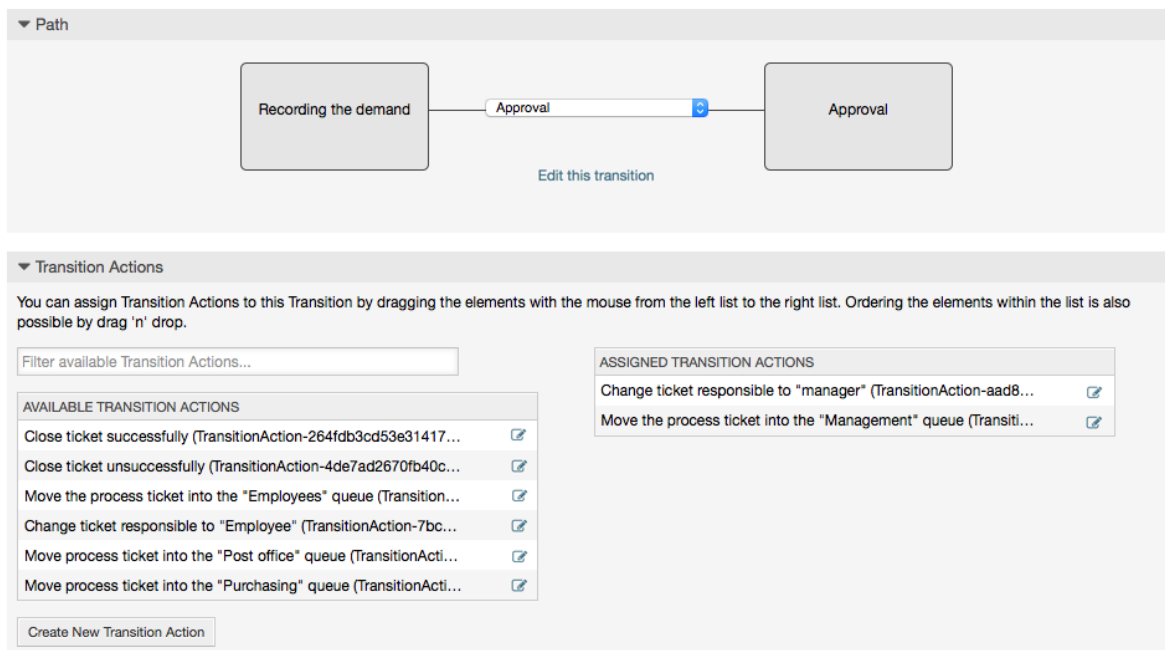
Now that the "Path" between the Actions is defined, then we need to assign the Transition Actions to the Transition, double click the Transition label (in the canvas), this will open a new popup window.

Abbildung 5.21. Verbinden Sie die Aktivitäten unter Verwendung von Übergängen



Drag the needed Transition Actions from Available Transition Actions pool and drop them into the Assigned Transition Actions pool and click on submit button.

Abbildung 5.22. Zuweisen von Übergangs-Aktionen



The screenshot shows the 'Path' section of a process editor. It displays a flow from 'Recording the demand' to 'Approval' via a transition labeled 'Approval'. Below the transition, there is an 'Edit this transition' link. The 'Transition Actions' section below shows a list of available actions and a list of assigned actions. The assigned actions are: 'Change ticket responsible to "manager" (TransitionAction-aad8...)' and 'Move the process ticket into the "Management" queue (Transiti...)'.

Then back in the main process edit screen click on save button below the canvas to save all other changes.

Complete the "path" adding the following Activities, Transitions and Transition Actions:

Aufnahme der Nachfrage bis zur "Approval"

- Starting point: Activity: "Recording the demand" (A1)
- Possible Transition: "Approval" (T1)
 - If the condition of this activity is fulfilled, the ticket will move to Activity: "Approval" (A2)
 - Zusätzlich werden die folgenden Übergangs-Aktionen ausgeführt:
 - "Verschiebe das Prozessticket in die Management-Queue" (TA1)
 - "Ändere den Ticket-Verantwortlichen zu 'Manager'" (TA2)

The Activity: "Recording the demand" (A1) is a defined step of the process ticket, where there is the possibility for the Transition: "Approval" (T1). If this applies, the ticket will move to the next Activity: "Approval" (A2), and the Transition Actions: "Move the process ticket into the 'Management' queue" (TA1) and "Change ticket responsible to 'manager'" (TA2) are executed. In the Activity: "Approval" (A2), the Activity Dialogs: "Approval denied" (AD2) and "Approved" (AD3) are available.

Genehmigung

- Startpunkt: Aktivität "Approval" (A2)
- Mögliche Übergänge:
 - "Genehmigung verweigert" (T2)
 - If this matches, the process ticket will move to Activity: "Process complete" (A5).
 - Zusätzlich werden die folgenden Übergangs-Aktionen ausgeführt:

- "Move process ticket into the 'Employees' queue" (TA3)
- Ändere den Ticket-Verantwortlichen zu 'Employee'" (TA4)
- Ticket erfolglos schließen" (TA8)
- "Genehmigt" (T3)
 - Wenn dies zutrifft, wird das Ticket in die Aktivität: "Order" (A3) geschoben.
 - Zusätzlich werden die folgenden Übergangs-Aktionen ausgeführt:
 - "Verschiebe das Prozessticket in die 'Purchasing'-Queue" (TA5)

We can see that from the current Activity, which defines a step of the process ticket, there are one or more possibilities for Transition which have exactly one target Activity (and possibly one or more Transition Actions).

Auftrag

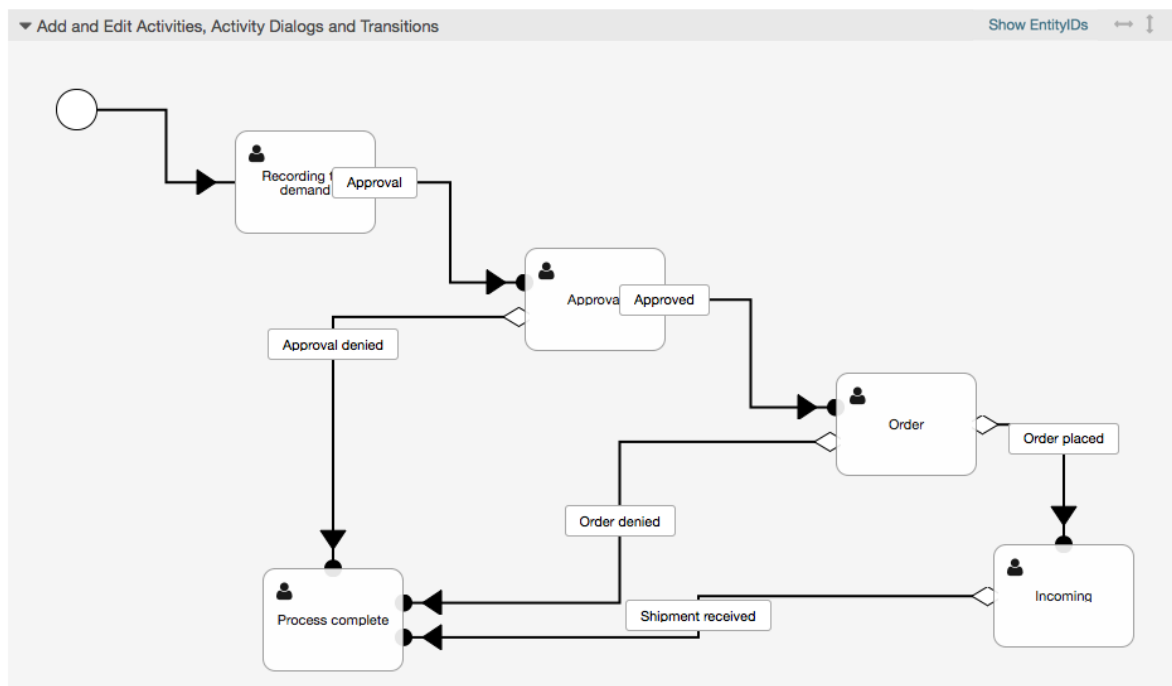
- Startpunkt: Aktivität "Order" (A3)
- Mögliche Übergänge:
 - "Bestellung abgelehnt" (T4)
 - If this matches, the process ticket will move to Activity: "Process complete" (A5).
 - Zusätzlich werden die folgenden Übergangs-Aktionen ausgeführt:
 - "Move process ticket into the 'Employees' queue" (TA3)
 - "Setze den Ticketverantwortlichen auf 'Employee'" (TA4)
 - Ticket erfolglos schließen" (TA8)
 - "Bestellung aufgegeben" (T5)
 - If this matches, the process ticket will move to Activity: "Incoming" (A4).
 - Zusätzlich werden die folgenden Übergangs-Aktionen ausgeführt:
 - "Verschiebe das Prozessticket in die 'Post office'-Queue" (TA6)

Eingehend

- Startpunkt: Aktivität "Incoming" (A4)
- Mögliche Übergänge:
 - "Sendung empfangen" (T6)
 - If this matches, the process ticket will move to Activity: "Process complete" (A5).
 - Zusätzlich werden die folgenden Übergangs-Aktionen ausgeführt:
 - "Move process ticket into the 'Employees' queue" (TA3)
 - "Setze den Ticketverantwortlichen auf 'Employee'" (TA4)
 - "Ticket erfolgreich schließen" (TA7)

Der komplette Pfad für den Buchbestellprozess schaut wie folgt aus:

Abbildung 5.23. vollständiger Buchbestellprozesspfad



After you finish the process path please click on "Save" button in the lower part of the canvas and then click on "Synchronize All Processes" button. This will gather all processes information from the Database and create a cache file (in Perl language). This cache file is actually the processes configuration that the system will use to create or use process tickets.

Any change that is made of the process (in the GUI) will require to re-synchronize the cache file in order to get the change reflected in the system.

It is also possible to import the whole process from a YAML file, but it is still necessary to create all Dynamic Fields, Users, Queues, etc that are needed by each process before the import.

Notice that if the process requires the use of ACLs those are also needed to be set manually.

The following is the complete YAML file for the book ordering process example:

```

---
Activities:
  A1:
    ActivityDialogs:
      - AD1
    ChangeTime: 2012-11-23 14:49:22
    Config:
      ActivityDialog:
        1: AD1
    CreateTime: 2012-11-23 11:49:38
    EntityID: A1
    ID: 151
    Name: Recording the demand
  A2:
    ActivityDialogs:
      - AD2
      - AD3
    ChangeTime: 2012-12-13 00:55:12
    Config:
      ActivityDialog:

```

```

    1: AD2
    2: AD3
  CreateTime: 2012-11-23 11:50:11
  EntityID: A2
  ID: 152
  Name: Approval
A3:
  ActivityDialogs:
    - AD4
    - AD5
  ChangeTime: 2012-11-23 18:12:14
  Config:
    ActivityDialog:
      1: AD4
      2: AD5
  CreateTime: 2012-11-23 11:50:35
  EntityID: A3
  ID: 153
  Name: Order
A4:
  ActivityDialogs:
    - AD6
  ChangeTime: 2012-11-23 18:12:35
  Config:
    ActivityDialog:
      1: AD6
  CreateTime: 2012-11-23 11:51:00
  EntityID: A4
  ID: 154
  Name: Incoming
A5:
  ActivityDialogs: []
  ChangeTime: 2012-11-23 11:51:33
  Config: {}
  CreateTime: 2012-11-23 11:51:33
  EntityID: A5
  ID: 155
  Name: Process complete
ActivityDialogs:
  AD1:
    ChangeTime: 2012-12-06 02:16:21
    Config:
      DescriptionLong: ''
      DescriptionShort: Recoding the demand
      FieldOrder:
        - DynamicField_Author
        - DynamicField_ISBN
        - DynamicField_Title
        - DynamicField_Status
      Fields:
        DynamicField_Author:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
        DynamicField_ISBN:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
        DynamicField_Status:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
        DynamicField_Title:
          DefaultValue: ''
          DescriptionLong: ''
          DescriptionShort: ''
          Display: 1
    Interface:
      - AgentInterface

```

```
Permission: ''
RequiredLock: 0
SubmitAdviceText: ''
SubmitButtonText: ''
CreateTime: 2012-11-23 14:34:43
EntityID: AD1
ID: 154
Name: Recording the demand
AD2:
ChangeTime: 2012-11-23 14:57:41
Config:
  DescriptionLong: ''
  DescriptionShort: Approval denied
  FieldOrder:
    - Article
    - DynamicField_Status
  Fields:
    Article:
      Config:
        ArticleType: note-internal
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
      DynamicField_Status:
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
    Interface:
      - AgentInterface
  Permission: ''
  RequiredLock: 0
  SubmitAdviceText: ''
  SubmitButtonText: Deny Request
CreateTime: 2012-11-23 14:36:39
EntityID: AD2
ID: 155
Name: Approval denied
AD3:
ChangeTime: 2012-12-14 03:14:23
Config:
  DescriptionLong: ''
  DescriptionShort: Approved
  FieldOrder:
    - DynamicField_Status
  Fields:
    DynamicField_Status:
      DefaultValue: ''
      DescriptionLong: ''
      DescriptionShort: ''
      Display: 1
    Interface:
      - AgentInterface
  Permission: ''
  RequiredLock: 0
  SubmitAdviceText: ''
  SubmitButtonText: Approve Request
CreateTime: 2012-11-23 14:37:35
EntityID: AD3
ID: 156
Name: Approved
AD4:
ChangeTime: 2012-11-23 14:58:52
Config:
  DescriptionLong: ''
  DescriptionShort: Order rejected
  FieldOrder:
    - Article
    - DynamicField_Status
  Fields:
    Article:
```

```
    Config:
      ArticleType: note-internal
      DefaultValue: ''
      DescriptionLong: ''
      DescriptionShort: ''
      Display: 1
    DynamicField_Status:
      DefaultValue: ''
      DescriptionLong: ''
      DescriptionShort: ''
      Display: 1
  Interface:
    - AgentInterface
  Permission: ''
  RequiredLock: 0
  SubmitAdviceText: ''
  SubmitButtonText: Reject Order
  CreateTime: 2012-11-23 14:38:48
  EntityID: AD4
  ID: 157
  Name: Order rejected
AD5:
  ChangeTime: 2012-12-06 02:20:12
  Config:
    DescriptionLong: ''
    DescriptionShort: Order placed
    FieldOrder:
      - DynamicField_DeliveryDate
      - DynamicField_Price
      - DynamicField_Supplier
      - DynamicField_Status
    Fields:
      DynamicField_DeliveryDate:
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
      DynamicField_Price:
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
      DynamicField_Status:
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
      DynamicField_Supplier:
        DefaultValue: ''
        DescriptionLong: ''
        DescriptionShort: ''
        Display: 1
  Interface:
    - AgentInterface
  Permission: ''
  RequiredLock: 0
  SubmitAdviceText: ''
  SubmitButtonText: Place Order
  CreateTime: 2012-11-23 14:41:28
  EntityID: AD5
  ID: 158
  Name: Order placed
AD6:
  ChangeTime: 2012-11-23 14:42:43
  Config:
    DescriptionLong: ''
    DescriptionShort: Shipment received
    FieldOrder:
      - DynamicField_DateOfReceipt
      - DynamicField_Status
    Fields:
      DynamicField_DateOfReceipt:
```

```

    DefaultValue: ''
    DescriptionLong: ''
    DescriptionShort: ''
    Display: 1
  DynamicField_Status:
    DefaultValue: ''
    DescriptionLong: ''
    DescriptionShort: ''
    Display: 1
  Interface:
    - AgentInterface
  Permission: ''
  RequiredLock: 0
  SubmitAdviceText: ''
  SubmitButtonText: ''
  CreateTime: 2012-11-23 14:42:43
  EntityID: AD6
  ID: 159
  Name: Shipment received
Process:
  Activities:
    - A1
    - A2
    - A3
    - A4
    - A5
  ChangeTime: 2012-12-06 02:31:59
  Config:
    Description: The process to order a book
    Path:
      A1:
        T1:
          ActivityEntityID: A2
          TransitionAction:
            - TA2
            - TA1
        A2:
          T2:
            ActivityEntityID: A5
            TransitionAction:
              - TA3
              - TA4
              - TA8
          T3:
            ActivityEntityID: A3
            TransitionAction:
              - TA5
        A3:
          T4:
            ActivityEntityID: A5
            TransitionAction:
              - TA3
              - TA4
              - TA8
          T5:
            ActivityEntityID: A4
            TransitionAction:
              - TA6
        A4:
          T6:
            ActivityEntityID: A5
            TransitionAction:
              - TA3
              - TA4
              - TA7
      A5: {}
    StartActivity: A1
    StartActivityDialog: AD1
  CreateTime: 2012-11-23 11:45:12
  EntityID: P1
  ID: 94
  Layout:

```



```
A1:
  left: 172
  top: 63
A2:
  left: 402
  top: 156
A3:
  left: 649
  top: 255
A4:
  left: 774
  top: 391
A5:
  left: 194
  top: 410
Name: Book ordering
State: Active
StateEntityID: S1
TransitionActions:
  - TA1
  - TA2
  - TA3
  - TA4
  - TA8
  - TA5
  - TA3
  - TA4
  - TA8
  - TA6
  - TA3
  - TA4
  - TA7
Transitions:
  - T1
  - T2
  - T3
  - T4
  - T5
  - T6
TransitionActions:
TA1:
  ChangeTime: 2012-11-23 16:01:37
  Config:
    Queue: Management
    Module: Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet
  CreateTime: 2012-11-23 15:50:59
  EntityID: TA1
  ID: 61
  Name: Move the process ticket into the "Management" queue
TA2:
  ChangeTime: 2012-11-23 16:02:12
  Config:
    Config:
      Responsible: manager
    Module: Kernel::System::ProcessManagement::TransitionAction::TicketResponsibleSet
  CreateTime: 2012-11-23 15:58:22
  EntityID: TA2
  ID: 62
  Name: Change ticket responsible to "manager"
TA3:
  ChangeTime: 2012-11-24 14:27:02
  Config:
    Config:
      Queue: Employees
    Module: Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet
  CreateTime: 2012-11-23 16:02:54
  EntityID: TA3
  ID: 63
  Name: Move the process ticket into the "Employees" queue
TA4:
  ChangeTime: 2012-11-23 16:04:06
```

```

Config:
  Config:
    Responsible: Employee
    Module: Kernel::System::ProcessManagement::TransitionAction::TicketResponsibleSet
  CreateTime: 2012-11-23 16:04:06
  EntityID: TA4
  ID: 64
  Name: Change ticket responsible to "Employee"
TA5:
  ChangeTime: 2012-12-06 02:18:34
  Config:
    Config:
      Queue: Purchasing
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet
    CreateTime: 2012-11-23 16:04:54
    EntityID: TA5
    ID: 65
    Name: Move process ticket into the "Purchasing" queue
TA6:
  ChangeTime: 2012-12-06 02:18:48
  Config:
    Config:
      Queue: Post office
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet
    CreateTime: 2012-11-23 16:06:20
    EntityID: TA6
    ID: 66
    Name: Move process ticket into the "Post office" queue
TA7:
  ChangeTime: 2012-12-06 02:29:55
  Config:
    Config:
      State: closed successful
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketStateSet
    CreateTime: 2012-12-06 02:29:27
    EntityID: TA7
    ID: 67
    Name: Close ticket successfully
TA8:
  ChangeTime: 2012-12-06 02:31:12
  Config:
    Config:
      State: closed unsuccessful
      Module: Kernel::System::ProcessManagement::TransitionAction::TicketStateSet
    CreateTime: 2012-12-06 02:31:12
    EntityID: TA8
    ID: 68
    Name: Close ticket unsuccessfully
Transitions:
T1:
  ChangeTime: 2012-11-23 15:12:20
  Config:
    Condition:
      1:
        Fields:
          DynamicField_Status:
            Match: Approval
            Type: String
          Type: and
        ConditionLinking: and
    CreateTime: 2012-11-23 11:53:52
    EntityID: T1
    ID: 94
    Name: Approval
T2:
  ChangeTime: 2012-11-23 15:12:50
  Config:
    Condition:
      1:
        Fields:
          DynamicField_Status:
            Match: Approval denied

```

```
    Type: String
  Type: and
  ConditionLinking: and
  CreateTime: 2012-11-23 11:54:26
  EntityID: T2
  ID: 95
  Name: Approval denied
T3:
  ChangeTime: 2012-11-23 15:13:29
  Config:
    Condition:
      1:
        Fields:
          DynamicField_Status:
            Match: Approved
            Type: String
          Type: and
        ConditionLinking: and
        CreateTime: 2012-11-23 11:54:54
        EntityID: T3
        ID: 96
        Name: Approved
T4:
  ChangeTime: 2012-11-23 15:14:08
  Config:
    Condition:
      1:
        Fields:
          DynamicField_Status:
            Match: Order denied
            Type: String
          Type: and
        ConditionLinking: and
        CreateTime: 2012-11-23 11:55:25
        EntityID: T4
        ID: 97
        Name: Order denied
T5:
  ChangeTime: 2012-11-23 18:30:33
  Config:
    Condition:
      1:
        Fields:
          DynamicField_Status:
            Match: Order placed
            Type: String
          Type: and
        ConditionLinking: and
        CreateTime: 2012-11-23 11:56:15
        EntityID: T5
        ID: 98
        Name: Order placed
T6:
  ChangeTime: 2012-11-23 15:15:30
  Config:
    Condition:
      1:
        Fields:
          DynamicField_Status:
            Match: Shipment received
            Type: String
          Type: and
        ConditionLinking: and
        CreateTime: 2012-11-23 11:56:48
        EntityID: T6
        ID: 99
        Name: Shipment received
```

2.4. Prozesskonfigurationsempfehlung

2.4.1. Prozess

Ein Prozess modelliert den Pfad eines Workflows/Prozess. Die Wegpunkte auf diesem Pfad können Aktivitäten oder Übergänge sein. Auf diese wird später noch genauer eingegangen.

2.4.1.1. Prozesskonfiguration

The Process configuration can be done in the file Kernel/Config.pm but it is strongly recommended to create new files like Kernel/Config/Files/MyProcess.pm. Notice that the GUI generates the file Kernel/Config/File/ZZZProcessManagement please avoid to use that filename, otherwise it will be overwritten when you sync processes. Let's see an example process configuration (from process cache file):

```
$Self->{'Process'} = {  
  'P1' => {  
    Name => 'Book order',  
    CreateTime => '16-02-2012 13:37:00',  
    CreateBy => '1',  
    ChangeTime => '17-02-2012 13:37:00',  
    ChangeBy => '1',  
    State => 'Active',  
    StartActivity => 'A1',  
    StartActivityDialog => 'AD1',  
    Path => {  
      'A1' => {  
        'T1' => {  
          ActivityEntityID => 'A2',  
        },  
      },  
      'A2' => {  
        'T2' => {  
          ActivityEntityID => 'A3',  
        },  
      },  
    },  
  },  
  'P2' => {  
    Name => 'IT order',  
    CreateTime => '26-02-2012 13:37:00',  
    CreateBy => '1',  
    ChangeTime => '27-02-2012 13:37:00',  
    ChangeBy => '1',  
    State => 'Active',  
    StartActivity => 'A2',  
    StartActivityDialog => 'AD2',  
    Path => {  
      'A2' => {  
        'T3' => {  
          ActivityEntityID => 'A4',  
        },  
      },  
    },  
  },  
};
```

2.4.1.2. Name

Der Name des Prozesses, dieser kann vom Agenten ausgewählt werden, wenn ein neues Prozessticket erstellt wird.

2.4.1.3. CreateTime

Die Zeit, wann der Prozess erstellt wurde.

2.4.1.4. CreateBy

Die UID des Users, der den Prozess erstellt hat.

2.4.1.5. ChangeTime

Die Zeit, wann der Prozess zuletzt geändert wurde.

2.4.1.6. ChangeBy

Die UID des Users, der den Prozess zuletzt bearbeitet hat.

2.4.1.7. State

Definiert den Status eines Prozesses. Möglich Werte sind:

- 'Active' sind alle Prozesse, welche in neuen Prozesstickets genutzt werden können.
- 'FadeAway' sind Prozesse, welche nicht mehr für neue Tickets genutzt werden können, aber bereits existierende Tickets können diesen Prozess immer noch ausführen.
- 'Inactive' Prozesse sind deaktiviert und können nicht neue und bereits existierende Tickets genutzt werden.

2.4.1.8. StartActivity

Wenn ein neues Prozessticket erstellt wird, muss eine Startaktivität definiert werden. Sobald das Ticket erstellt wurde, wird diese Aktivität gesetzt und als Basis für die erste Prüfung des Übergangs genutzt.

2.4.1.9. StartActivityDialog

Für neue Prozesstickets muss ein StartActivityDialog definiert werden. Dieser wird angezeigt, wenn ein neues Prozessticket erstellt wird (nachdem der Prozess ausgewählt wurde). Das Ticket existiert noch nicht, wird aber nach Übermittlung des StartActivityDialog erstellt.

2.4.1.10. Pfad

Der Pfad beinhaltet die aktuelle Prozessstruktur der Aktivitäten und möglichen Übergänge zwischen ihnen.

```
'A1' => {  
'T1' => {  
ActivityEntityID => 'A2',  
},  
'T2' => {  
ActivityEntityID => 'A3',  
},  
'T3' => {  
ActivityEntityID => 'A4',  
TransitionAction => ['TA1', 'TA2'],  
},  
},
```

Wenn ein Prozessticket in Aktivität 'A1' ist, hat es 3 mögliche Wege um zu einer anderen Aktivität zugelangt. In den Übergängen 'T1' bis 'T3' wurden Bedingungen festgelegt, die ein Prozessticket erfüllen muss um zu einer anderen Aktivität zu gelangen.

If in this case all the values of the process ticket and its dynamic fields that are needed for the Transition 'T2' are correct, the ticket will be moved from Activity 'A1' to 'A3'. After

an ActivityDialog is submitted, or any other change is made to a ticket, it will be checked for possible Transitions from the current Activity. If multiple Transitions are possible, the first one will be used (based on numerical sorting of the TransitionIDs).

Additionally, it is possible to assign Transition Actions to Transitions in the Path configuration. These are modules which are executed after a successful Transition. They have to be specified in array form as in the example, we'll talk about the details later.

2.4.2. Aktivität

An Activity contains one or more Activity Dialogs and models a 'step' in the process. All Activity Dialogs of the current Activity are displayed in the ticket zoom and can be used until the conditions of a Transition are fulfilled.

2.4.2.1. Aktivitäts-Konfiguration

Hier sehen Sie eine Beispiel Aktivitätskonfiguration:

```
$Self->{'Process::Activity'} =  
{  
  'A1' => {  
    Name => 'Activity 1 optional',  
    CreateTime => '16-02-2012 13:37:00',  
    CreateBy => '1',  
    ChangeTime => '17-02-2012 13:37:00',  
    ChangeBy => '1',  
    ActivityDialog => {  
      1 => 'AD1',  
    },  
  },  
  'A2' => {  
    Name => 'Activity 2 optional',  
    CreateTime => '16-02-2012 13:37:00',  
    CreateBy => '1',  
    ChangeTime => '17-02-2012 13:37:00',  
    ChangeBy => '1',  
    ActivityDialog => {  
      1 => 'AD5',  
      2 => 'AD6',  
      3 => 'AD1',  
    },  
  },  
};
```

2.4.2.2. Name

Der Name der Aktivität.

2.4.2.3. CreateTime

Die Erstellzeit.

2.4.2.4. CreateBy

Die UID des Benutzers, der die Aktivität erstellt hat.

2.4.2.5. ChangeTime

Die letzte Änderung.

2.4.2.6. ChangeBy

Die UID des Benutzers, der die letzte Aktivitätsänderung gemacht hat.

2.4.2.7. Aktivitäts-Dialog

Activity Dialog contains the list of Activity Dialogs which are available in this Activity. All Activity Dialogs of the current Activity are displayed in the ticket zoom. Their order is set by the order in the configuration, here 'AD5' is shown before 'AD6' and 'AD1'.

2.4.3. Aktivitäts-Dialog

Ein Aktivitätsdialog ist eine besondere Oberfläche und kann in verschiedenen Aktivitäten wiederverwendet werden.

2.4.3.1. Aktivitätsdialog-Konfiguration

Let's see an example config:

```
$Self->{'Process::ActivityDialog'} = {
  'AD1' => {
    Name           => 'ActivityDialog 1 optional',
    DescriptionShort => 'Basic info',
    DescriptionLong => 'Please insert the necesesary basic information for IT orders',
    CreateTime     => '28-02-2012 13:37:00',
    CreateBy       => '1',
    ChangeTime     => '29-02-2012 13:37:00',
    ChangeBy       => '1',
    Fields => {
      PriorityID => {
        DescriptionShort => 'Priority ID',
        DescriptionLong  => 'Enter the priority here',
        Display           => 2,
      },
    },
    FieldOrder     => [ 'PriorityID' ],
    SubmitAdviceText => 'Note: If you submit the form...',
    SubmitButtonText => 'Send request',
  },
  'AD2' => {
    Name           => 'ActivityDialog 2 optional',
    DescriptionShort => 'Basic info',
    DescriptionLong => 'Please insert the necesesary basic information for Book
orders',
    CreateTime     => '28-02-2012 13:37:00',
    CreateBy       => '1',
    ChangeTime     => '29-02-2012 13:37:00',
    ChangeBy       => '1',
    Fields => {
      StateID => {
        DescriptionShort => 'State ID',
        DescriptionLong  => 'Enter the state here',
        Display           => 2,
        DefaultValue     => '2',
      },
      Queue => {
        DescriptionShort => 'Queue ID',
        DescriptionLong  => 'Enter the queue here',
        Display           => 2,
        DefaultValue     => 'Raw',
      },
      Title => {
        DescriptionShort => 'Title',
        DescriptionLong  => 'Enter the title here',
        Display           => 1,
        DefaultValue     => 'Default Title',
      },
    },
    DynamicField_Anzahl => {
      DescriptionShort => 'Amount',
      DescriptionLong  => 'Enter the amount here',
      Display           => 2,
      DefaultValue     => '4',
    },
  },
}
```

```

    },
  },
  FieldOrder      => [ 'DynamicField_Anzahl', 'StateID', 'Queue', 'Title' ],
  SubmitAdviceText => 'Note: If you submit the form...',
  SubmitButtonText => 'Send request',
  },
};

```

2.4.3.2. Name

Name des Aktivitäts-Dialog.

2.4.3.3. CreateTime

Die Erstellzeit.

2.4.3.4. CreateBy

Die UID des Benutzers, der den Aktivitätendialog erstellt hat.

2.4.3.5. ChangeTime

Die letzte Änderung.

2.4.3.6. ChangeBy

Die UID des Benutzers, der die letzte Änderung an diesem Aktivitätsdialog gemacht hat.

2.4.3.7. Felder

Beinhaltet alle Felder, welche in diesem Aktivitätsdialog angezeigt werden können. Die folgenden Felder können zurzeit benutzt werden:

```

Title
State
StateID
Priority
PriorityID
Lock
LockID
Queue
QueueID
Customer
CustomerID
CustomerNo
CustomerUserID
Owner
OwnerID
Type
TypeID
SLA
SLAID
Service
ServiceID
Responsible
ResponsibleID
PendingTime
DynamicField_$FieldName # for all dynamic fields

```

Beispielkonfiguration eines einfachen Feldes:

```

StateID => {
DescriptionShort => 'State ID',

```



```
DescriptionLong => 'Enter the state here',
Display => 2,
DefaultValue => '2',
},
```

The field "Article" is a special case. If it is present in a "Fields" configuration, the Activity Dialog will contain a complete Richtext editor with subject field and attachment handling. The entered text will then be added to the ticket as an article and sent by email. Let's see an example Article field configuration:

```
Article => {
DescriptionShort => 'Please insert your comment here.',
DescriptionLong => '',
Display => 1,
Config => {
ArticleType => 'note-internal',
LabelSubject => '',
LabelBody => '',
},
},
```

Werfen wir einen Blick auf die Optionen der Konfigurationsfelder:

2.4.3.7.1. Beschreibung (kurz)

Optionale Kurzbeschreibung, welche im Titel des Feldes angezeigt wird.

2.4.3.7.2. Beschreibung (lang)

Optionale längere Beschreibung, welche bei MouseOver über dem Feld angezeigt wird. Zum Beispiel Hinweise, wie das Feld ausgefüllt werden soll.

2.4.3.7.3. Anzeige

Steuert, ob das Feld angezeigt und/oder verpflichtend ist. Möglich Werte:

- '0': Das Feld ist unsichtbar. Das kann hilfreich sein, wenn Feld-Werte automatisch gesetzt werden sollen. Der konfigurierte Standard-Wert wird in diesem Fall gespeichert.
- '1': Das Feld ist sichtbar, aber optional.
- '2': field is visible and mandatory. The following fields can only be invisible or mandatory:

```
QueueID
Queue
State
StateID
Lock
LockID
Priority
PriorityID
Type
TypeID
```

Wenn Felder als optional konfiguriert sind und kein Wert vom Benutzer eingegeben wurde, wird der Standard-Wert übernommen, wenn der Aktivitätsdialog vom Benutzer übermittelt wurde.

2.4.3.7.4. Standard-Wert

For fields with 'ID' (like QueueID, OwnerID), this refers to the database ID of the value. For other fields without 'ID' (like Queue, Owner), the DefaultValue must contain the value itself. Example:

```
Queue => {
  DescriptionShort => 'Queue',
  DescriptionLong => 'Hier die Queue eingeben',
  Display => 2,
  DefaultValue => 'Raw',
},
```

2.4.3.8. Feldreihenfolge

Here the display order of the fields is configured. IMPORTANT: Invisible fields also must be configured here, because only configured fields will be considered when saving. Fields which are not configured will not be saved.

2.4.3.9. SubmitAdviceText

Optionaler Text, der direkt über der Übermitteln-Schaltfläche angezeigt wird und zusätzliche Hilfestellungen oder Ratschläge gibt.

2.4.3.10. SubmitButtonText

Optionaler Text für den "Übermitteln" Button

2.4.4. Übergang

Ein Übergang entscheidet - basierend auf konfigurierbaren Bedingungen - welchen Pfad der Prozess nimmt, zum Beispiel zu welcher Aktivität das Prozessticket spring.

2.4.4.1. Übergangs-Konfiguration

Hier sehen Sie ein Beispiel:

```
$Self->{'Process::Transition'} = {
  'T1' => {
    Name => 'Transition 1',
    CreateTime => '14-03-2012 13:37:00', # optional
    CreateBy => '1', # optional
    ChangeTime => '15-03-2012 13:37:00', # optional
    ChangeBy => '15-03-2012 13:37:00', # optional
    Condition => {
      Cond1 => {
        Fields => {
          StateID => {
            Type => 'String',
            Match => '1',
          },
        },
      },
    },
  },
  'T2' => {
    Name => 'Transition 2 optional',
    CreateTime => 'DATE', # optional
    CreateBy => 'USERID', # optional
    ChangeTime => 'DATE', # optional
    ChangeBy => 'USERID', # optional
    Condition => {
      Cond1 => {
        Queue => 'Raw',
        DynamicField_Farbe => '2',
        DynamicField_Anzahl => '1',
      },
    },
  },
};
```

2.4.4.2. Name

Name des Übergangs.

2.4.4.3. CreateTime

Die Zeit, wann er erstellt wurde.

2.4.4.4. CreateBy

Die UID des Benutzers, der den Übergang erstellt hat.

2.4.4.5. ChangeTime

Die letzte Änderung

2.4.4.6. ChangeBy

UID des Users, der den Übergang geändert hat.

2.4.4.7. Bedingung

Beinhaltet alle Bedingungen, die notwendig sind, damit der Übergang wirksam ist. Beispiel:

```
Condition => {
  Type => 'and',
  Cond1 => {
    Type => 'and',
    Fields => {
      StateID => {
        Type => 'String',
        Match => '1',
      },
      DynamicField_Marke => {
        Type => 'String',
        Match => 'VW',
      },
    },
  },
  Cond2 => {
    Type => 'and',
    Fields => {
      Queue => {
        Type => 'String',
        Match => 'Raw',
      },
    },
  },
},
```

Werfen wir einen Blick auf den Zustand der Konfiguration im Detail.

2.4.4.7.1. Typ (Bedingung)

Specifies the way the different condition elements are connected to each other. Possible values:

- 'UND': Das ist der Standard. Alle Bedingungen müssen erfüllt sein damit der Übergang angewendet wird.
- 'ODER': Mindestens eine Bedingung muss erfüllt sein.
- 'XOR': Genau eine Bedingung muss erfüllt sein, nicht mehr.

2.4.4.7.2. Cond1

This is the name of an example condition. It can be freely chosen. Conditions are evaluated in sorted order.

2.4.4.7.3. Typ (Cond)

Specifies the way how the individual field tests of this condition are connected to each other. Possible values:

- 'UND': Das ist der Standard. Alle Feldtests müssen erfüllt sein damit die Bedingung erfüllt ist.
- 'ODER': Mindestens ein Feldtest muss erfüllt sein.
- 'XOR': Genau ein Feldtest muss erfüllt sein, nicht mehr.

2.4.4.7.4. Felder

Specifies the particular fields whose values should be tested. From our example:

```
Fields => {  
  StateID => {  
    Type => 'String',  
    Match => '1',  
  },  
}
```

2.4.4.7.5. StateID

Example of a field name. The following ticket fields can be used:

```
Title  
State  
StateID  
Priority  
PriorityID  
Lock  
LockID  
Queue  
QueueID  
Customer  
CustomerID  
CustomerNo  
CustomerUserID  
Owner  
OwnerID  
Type  
TypeID  
SLA  
SLAID  
Service  
ServiceID  
Responsible  
ResponsibleID  
DynamicField_${FieldName} # for all DynamicFields
```

Beim Testen eines Feldes mit "ID" (wie SLAID), wird die Datenbank-ID des Feldes für die Prüfung verwendet werden, für andere Bereiche (wie SLA) wird der Ist-Wert für den Test verwendet.

2.4.4.7.6. Typ

Bestimmt die Art der Feldtests. Mögliche Werte:

- 'String': Vergleicht den Feldwert mit der in 'Match' angegebene Zeichenfolge. Stimmt überein, wenn sie genau gleich sind.
- 'Hash': Vergleicht den Feldwert (Hash) mit dem in 'Match' angegebenen Hash. Alle Hash-Werte müssen identisch sein.
- 'Array': Vergleicht den Wert des Feldes (Array) mit dem in 'Match' angegebene Array. Beide Listen müssen gleich sein.
- 'Regex': The field value can be tested with a regular expression. It is important that 'Match' contains *qr{ }xms* as a base condition. Between the braces the actual regular expression can be noted.
- 'Module': Allows you to use a perl module for condition checking. If it returns 1, the check was positive. You can find an example module in `Kernel/System/ProcessManagement/TransitionValidation/ValidateDemo.pm`.

2.4.5. Übergangs-Aktionen

Transition Actions are actions which can be triggered after successfully applied transitions (when a process ticket moves from one activity to another). These Transition Actions can be used to perform different changes on the ticket, e. g. change the Queue or the Owner of the ticket, and you can also create your own Transition Actions to perform other complex changes.

2.4.5.1. Übergangs-Aktionen-Konfiguration

Hier sehen Sie ein Beispiel:

```
$Self->{'Process::TransitionAction'} = {  
    'TA1' => {  
        Name => 'Queue Move',  
        Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet',  
        Config => {  
            Queue => 'Junk',  
            UserID => 123,  
        },  
    },  
};
```

2.4.5.2. Name

Name der Übergangs-Aktion

2.4.5.3. Modul

Spezifiziert das Perl-Modul, welches benutzt werden soll.

2.4.5.4. Konfiguration

This parameter contains all settings which are required for the module. Its content depends on the particular Transition Action module which is used. Please see the documentation of the individual modules for details. In our example, only the Queue must be specified. Nevertheless we are also sending UserID parameter, by using the UserID parameter. The transition action will be executed impersonating the user with the given UserID.

The use of UserID inside the "Config" parameter of a Transition Action is accepted by all Transition Actions (since OTRS 3.2.4). In this example it could be particularly important if the user that triggers the Transition does not have permissions to move the ticket to the queue 'Junk', while the user with the UserID 123 might have.

2.4.5.5. Wiederverwendung von Übergangs-Aktionsmodulen

To use Transition Action modules multiple times, just specify several Transition Actions in your configuration. Example:

```
$Self->{'Process::TransitionAction'} = {
  'TA1' => {
    Name => 'Queue Move Junk',
    Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet',
    Config => {
      Queue => 'Junk',
    },
  },
  'TA2' => {
    Name => 'Queue Move Raw',
    Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet',
    Config => {
      Queue => 'Raw',
    },
  },
};
```

Here the same module is used to move a process ticket into the 'Raw' queue, and another time to move it into the junk queue. The Transition Action which must be used for a particular Transition is determined from the 'Path' setting of the Process configuration.

2.4.5.6. Verfügbare Übergangs-Aktionen

OTRS comes with several Transition Actions that can be used in your processes. Here you can find their documentation and how they need to be configured.

2.4.5.6.1. DynamicFieldSet

Fügt eins oder mehrere dynamischen Felder an ein Prozessticket an. Beispiel:

```
$Self->{'Process::TransitionAction'} = {
  'TA1' => {
    Name => 'Set DynamicField MasterSlave to Master and Approved to 1',
    Module => 'Kernel::System::ProcessManagement::TransitionAction::DynamicFieldSet',
    Config => {
      MasterSlave => 'Master',
      Approved => '1',
    },
  },
};
```

'Name' definiert den Namen der konfigurierten Übergangsaktion.

'MasterSlave' and 'Approved' are given as examples of DynamicField names. The values of the fields ('Master' and '1') will be set by this TransitionAction.

2.4.5.6.2. TicketArticleCreate

Erzeugt einen Artikel und wird verwendet um Notizen oder E-Mail Antworten erzeugen. Beispiel:

```
$Self->{'Process::TransitionAction'} = {
  'TA1' => {
    Name => 'Article Create Note Internal',
    Module =>
    'Kernel::System::ProcessManagement::TransitionAction::TicketArticleCreate',
  },
};
```

```

    Config => {
      ArticleType      => 'note-internal',                #
note-external|phone|fax|sms|...                          #
excluding any email type
      SenderType      => 'agent',                        #
agent|system|customer
      ContentType     => 'text/plain; charset=ISO-8859-15', # or
optional Charset & MimeType (e.g. 'text/html; charset=UTF-8')
      Subject        => 'some short description',        #
required
      Body           => 'the message text',              #
required
      HistoryType     => 'OwnerUpdate',                  #
EmailCustomer|Move|AddNote|PriorityUpdate|WebRequestCustomer|...
      HistoryComment  => 'Some free text!',              #
      From           => 'Some Agent <email@example.com>', #
not required but useful
      To             => 'Some Customer A <customer-a@example.com>', #
not required but useful
      Cc            => 'Some Customer B <customer-b@example.com>', #
not required but useful
      ReplyTo       => 'Some Customer B <customer-b@example.com>', #
not required
      InReplyTo     => '<asdasdasd.12@example.com>',      #
not required but useful
      References    => '<asdasdasd.1@example.com> <asdasdasd.12@example.com>', #
not required but useful
      NoAgentNotify  => 0,                               # if
you don't want to send agent notifications
      AutoResponseType => 'auto reply',                 #
auto reject|auto follow up|auto reply/new ticket|auto remove

      ForceNotificationToUserID => [ 1, 43, 56 ],        # if
you want to force somebody
      ExcludeNotificationToUserID => [ 43, 56 ],
      # if you want full exclude somebody from notifications,
      # will also be removed in To: line of article,
      # higher prio as ForceNotificationToUserID
      ExcludeMuteNotificationToUserID => [ 43, 56 ],
      # the same as ExcludeNotificationToUserID but only the
      # sending gets muted, agent will still shown in To:
      # line of article
    },
  },
};

```

'Name' specifies the name of the configured TransitionAction. It can be freely chosen, but should reflect the purpose of the configured action.

'ArticleType' definiert den Typ des Artikels, der erstellt werden soll. Mögliche Werte: phone, fax, sms, webrequest, note-internal, note-external and note-report.

'SenderType' defines the sender type of the article. Possible values: agent, system, customer.

'ContentType' definiert den Inhaltstyp des Artikels. Mögliche Werte: 'text/plain; charset=ISO-8859-15' or any other valid charset and mime type.

'Betreff' definiert den Artikeltitel. Verpflichtend.

'Body' definiert den Artikelinhalt. Verpflichtend

'HistoryType' definiert den Typ des Historien-Eintrages. Möglich Werte: AddNote, ArchiveFlagUpdate, Bounce, CustomerUpdate, EmailAgent, EmailCustomer, EscalationResponseTimeNotifyBefore, EscalationResponseTimeStart, EscalationResponseTimeStop, EscalationSolutionTimeNotifyBefore, EscalationSolutionTimeStart, EscalationSolutionTimeStop, EscalationUpdateTimeNotifyBefore, EscalationUpdateTimeStart, EscalationUp-

dateTimeStop, FollowUp, Forward, Lock, LoopProtection, Merged, Misc, Move, NewTicket, OwnerUpdate, PhoneCallAgent, PhoneCallCustomer, PriorityUpdate, Remove, ResponsibleUpdate, SendAgentNotification, SendAnswer, SendAutoFollowUp, SendAutoReject, SendAutoReply, SendCustomerNotification, ServiceUpdate, SetPendingTime, SLAUpdate, StateUpdate, Subscribe, SystemRequest, TicketDynamicFieldUpdate, TicketLinkAdd, TicketLinkDelete, TimeAccounting, TypeUpdate, Unlock, Unsubscribe, WebRequestCustomer.

'HistoryContent' definiert den Inhalt von Historieneinträgen.

'Von', 'An', 'Cc' und 'ReplyTo' nehmen E-Mailadressen in der oben angegebenen Schreibweise.

'InReplyTo' und 'References' nehmen E-Mailnachrichten-IDs

'NoAgentNotify' - Ist dieser Wert auf 1 gesetzt, werden keine E-Mailbenachrichtigungen an Agenten verschickt.

'AutoResponseType' kann die folgenden Werte annehmen: Auto FollowUp, Auto-Reject, Auto-Remove, Auto-Reply, Auto-Reply/New Ticket.

'ForceNotificationToUserID', 'ExcludeNotificationToUserID', 'ExcludeMuteNotificationToUserID' können eine Liste verschiedener User beinhalten, die entweder immer benachrichtigt, nie benachrichtigt oder als benachrichtigt aufgeführt werden, aber keine Benachrichtigungs-E-Mail erhalten.

2.4.5.6.3. TicketCreate

Erstellt ein Ticket mit einem Artikel, das neue Ticket kann mit einem Prozessticket verlinkt werden. Beispiel:

```
$Self->{'Process::TransitionAction'} = {
  'TA1' => {
    Name => 'Ticket Create',
    Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketCreate',
    Config => {

      # ticket required:
      Title => 'Some Ticket Title',
      Queue => 'Raw', # or QueueID => 123,
      Lock => 'unlock',
      Priority => '3 normal', # or PriorityID => 2,
      State => 'new', # or StateID => 5,
      CustomerID => '123465',
      CustomerUser => 'customer@example.com',
      OwnerID => 123,

      # ticket optional:
      TN => $TicketObject->TicketCreateNumber(), # optional
      Type => 'Incident', # or TypeID => 1, not required
      Service => 'Service A', # or ServiceID => 1, not required
      SLA => 'SLA A', # or SLAID => 1, not required
      ResponsibleID => 123, # not required
      ArchiveFlag => 'y', # (y/n) not required
      PendingTime => '2011-12-23 23:05:00', # optional (for pending states)
      PendingTimeDiff => 123, # optional (for pending states)

      # article required:
      ArticleType => 'note-internal', # note-external|
phone|fax|sms|... # excluding any
      email type
      SenderType => 'agent', # agent|system|
customer
      ContentType => 'text/plain; charset=ISO-8859-15', # or optional
      Charset & MimeType (e.g. 'text/html; charset=UTF-8')
```



```

Subject      => 'some short description',          # required
Body         => 'the message text',                # required
HistoryType  => 'OwnerUpdate',                    #
EmailCustomer|Move|AddNote|PriorityUpdate|WebRequestCustomer|...
HistoryComment => 'Some free text!',

# article optional:
useful      From      => 'Some Agent <email@example.com>',      # not required but
but useful  To        => 'Some Customer A <customer-a@example.com>', # not required
but useful  Cc        => 'Some Customer B <customer-b@example.com>', # not required
useful      ReplyTo   => 'Some Customer B <customer-b@example.com>', # not required
useful      MessageID => '<asdasdasd.123@example.com>',          # not required but
useful      InReplyTo => '<asdasdasd.12@example.com>',          # not required but
not required but useful  References => '<asdasdasd.1@example.com> <asdasdasd.12@example.com>', #
to send agent notifications  NoAgentNotify => 0,          # if you don't want
follow up|auto reply/new ticket|auto remove  AutoResponseType => 'auto reply' # auto reject|auto
ForceNotificationToUserID => [ 1, 43, 56 ],          # if you want to
force somebody  ExcludeNotificationToUserID => [ 43,56 ],      # if you want full
exclude somebody from notifications,          # will also be
removed in To: line of article,              # higher prio as
ForceNotificationToUserID  ExcludeMuteNotificationToUserID => [ 43,56 ], # the same as
ExcludeNotificationToUserID but only the     # sending gets
muted, agent will still shown in To:        # line of article

TimeUnit      => 123

# other:
DynamicField_NameX => $Value,
LinkAs => $LinkType,          # Normal, Parent,
Child, etc. (respective original ticket)
UserID => 123,                # optional, to
override the UserID from the logged user
    },
};

```

'Name' specifies the name of the configured TransitionAction. It can be freely chosen, but should reflect the purpose of the configured action.

'Title' Der Ticket-Titel

'Queue' oder 'QueueID' gibt den Namen oder die ID der Queue an, in der das neue Ticket erstellt werden soll

'Lock' oder 'LockID' setzt den "gesperrt"-Status des Tickets.

'Priority' or 'PriorityID' specifies the name or id of the priority to be used in the new ticket.

'State' oder 'StateID' gibt den Namen oder die ID des Status des neuen Tickets an.

'CustomerID', die Kundennummer die im neuen Ticket gesetzt werden soll.

'CustomerUser', der Login des Kunden der dem Ticket zugewiesen wird.

'OwnerID', die ID des Agenten, dem das neue Ticket gehören soll.

'TN', eigene Nummer für das neue Ticket.

'Type' oder 'TypeID' gibt den Namen oder die ID des Tickettyps an, der im neuen Ticket verwendet werden soll.

'Service' oder 'ServiceID' gibt den Namen oder die ID des Services an, der im neuen Ticket verwendet wird.

'SLA' or 'SLAID' specifies the name or id of the SLA to be used in the new ticket.

'ResponsibleID', die ID des Agenten, der Verantwortlich für das neue Ticket ist.

'PendingTime', a predefined date to set the Ticket Pending Times, when the ticket state belongs to a pending state type.

'PendingTimeDiff', a dynamically date (expressed in seconds from current date/time) to set the Ticket Pending Times, when the ticket state belongs to a pending state type.

'ArticleType' definiert den Typ des Artikels, der erstellt werden soll. Mögliche Werte: phone, fax, sms, webrequest, note-internal, note-external and note-report.

SenderType definiert den Absendertyp des Artikel. Mögliche Werte sind: Agent, System, Customer.

'ContentType' definiert den Inhaltstyp des Artikels. Mögliche Werte: 'text/plain; charset=ISO-8859-15' or any other valid charset and mime type.

'Betreff' definiert den Artikeltitle. Verpflichtend.

'Body' definiert den Artikelinhalt. Verpflichtend

'HistoryType' definiert den Typ des Historien-Eintrages. Möglich Werte: AddNote, ArchiveFlagUpdate, Bounce, CustomerUpdate, EmailAgent, EmailCustomer, EscalationResponseTimeNotifyBefore, EscalationResponseTimeStart, EscalationResponseTimeStop, EscalationSolutionTimeNotifyBefore, EscalationSolutionTimeStart, EscalationSolutionTimeStop, EscalationUpdateTimeNotifyBefore, EscalationUpdateTimeStart, EscalationUpdateTimeStop, FollowUp, Forward, Lock, LoopProtection, Merged, Misc, Move, NewTicket, OwnerUpdate, PhoneCallAgent, PhoneCallCustomer, PriorityUpdate, Remove, ResponsibleUpdate, SendAgentNotification, SendAnswer, SendAutoFollowUp, SendAutoReject, SendAutoReply, SendCustomerNotification, ServiceUpdate, SetPendingTime, SLAUpdate, StateUpdate, Subscribe, SystemRequest, TicketDynamicFieldUpdate, TicketLinkAdd, TicketLinkDelete, TimeAccounting, TypeUpdate, Unlock, Unsubscribe, WebRequestCustomer.

'HistoryContent' definiert den Inhalt von Historieneinträgen.

'Von', 'An', 'Cc' und 'ReplyTo' nehmen E-Mailadressen in der oben angegebenen Schreibweise.

'InReplyTo' und 'References' nehmen E-Mailnachrichten-IDs

'NoAgentNotify' - Ist dieser Wert auf 1 gesetzt, werden keine E-Mailbenachrichtigungen an Agenten verschickt.

'AutoResponseType' kann die folgenden Werte annehmen: Auto FollowUp, Auto-Reject, Auto-Remove, Auto-Reply, Auto-Reply/New Ticket.

'ForceNotificationToUserID', 'ExcludeNotificationToUserID', 'ExcludeMuteNotificationToUserID' können eine Liste verschiedener User beinhalten, die entweder immer benachrichtigt, nie benachrichtigt oder als benachrichtigt aufgeführt werden, aber keine Benachrichtigungs-E-Mail erhalten.

'TimeUnit', ist die investierte Zeit in den aktuellen Ticketartikel die in Sekunden, Minuten, Stunden, etc. ausgedrückt wird.

'DynamicField_NameX' where DynamicField_ is a required prefix and NameX is the name of a Dynamic Field to be set in the new ticket (on ticket level, not article levels).

'LinkAs' to define the new ticket relation with originator ticket, from the new ticket point of view, for example Normal, Parent, Child etc.

2.4.5.6.4. TicketCustomerSet

Legt den Kunden für ein Prozessticket fest. Beispiel:

```
$Self->{'Process::TransitionAction'} = {
  'TA1' => {
    Name => 'Customer Set Customer to test',
    Module => 'Kernel::System::Process::TransitionAction::TicketCustomerSet',
    Config => {
      No => 'test',
      User => 'client-user-123',
      # or in other words
      # CustomerID => 'client123',
      # CustomerUserID => 'client-user-123',
    },
  },
};
```

'Name' definiert den Namen der konfigurierten Übergangsaktion.

'No' or 'CustomerID' set the Customer ID of the customer.

'User' or 'CustomerUserID' set the Username of the customer.

2.4.5.6.5. TicketLockSet

Ändert die Sperre eines Prozessticket. Beispiel:

```
$Self->{'Process::TransitionAction'} = {
  'TA1' => {
    Name => 'Set Lock to lock',
    Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketLockSet',
    Config => {
      Lock => 'lock',
      # or
      LockID => 2,
    },
  },
};
```

'Name' definiert den Namen der konfigurierten Übergangsaktion.

'Lock' definiert den neuen Sperrstatus des Prozesstickets.

'LockID' definiert die interne ID des neuen Sperrstatus.

2.4.5.6.6. TicketOwnerSet

Ändert den Besitzer eines Prozesstickets. Beispiel:

```
$Self->{'Process::TransitionAction'} = {
  'TA1' => {
    Name => 'Owner Set root@localhost',
```

```
Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketOwnerSet',
Config => {
Owner => 'root@localhost',
# or
OwnerID => 1,
},
},
};
```

'Name' definiert den Namen der konfigurierten Übergangsaktion.

'Owner' definiert den Loginnamen des neuen Besitzers.

'OwnerID' definiert die interne ID des neuen Besitzers.

2.4.5.6.7. TicketQueueSet

Verschiebt das Ticket in eine Ziel-Queue. Beispiel:

```
$Self->{'Process::TransitionAction'} = {
'TA1' => {
Name => 'Queue Move Raw',
Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketQueueSet',
Config => {
Queue => 'Raw',
# or
# QueueID => '2',
},
},
};
```

'Name' definiert den Namen der konfigurierten Übergangsaktion.

'Queue' definiert den Namen der Ziel-Queue.

'QueueID' definiert die interne ID der Ziel-Queue.

2.4.5.6.8. TicketResponsibleSet

Ändert den Verantwortlichen eines Prozesstickets. Beispiel:

```
$Self->{'Process::TransitionAction'} = {
'TA1' => {
Name => 'Responsible Set root@localhost',
Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketResponsibleSet',
Config => {
Responsible => 'root@localhost',
# or
ResponsibleID => 1,
},
},
};
```

'Name' definiert den Namen der konfigurierten Übergangsaktion.

'Responsible' definiert den Loginnamen des neuen Verantwortlichen

'ResponsibleID' definiert die interne ID des neuen Verantwortlichen.

2.4.5.6.9. TicketServiceSet

Ordnet einem Prozessticket einen Service zu. Das Ticket benötigt einen Kunden und der Service muss diesem Kunden zugeordnet sein. Beispiel:

```
$Self->{'Process::TransitionAction'} = {
  'TA1' => {
    Name => 'Set MyService service',
    Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketServiceSet',
    Config => {
      Service => 'MyService',
      # or
      ServiceID => 123,
    },
  },
};
```

'Name' definiert den Namen der konfigurierten Übergangsaktion.

'Service' definiert den neuen Service des Prozesstickets. Es wird der vollständige Name des Service benötigt. (zum Beispiel: GrandFatherService::FatherService::SonService).

'ServiceID' definiert die interne ID des neuen Services.

2.4.5.6.10. TicketSLASet

Ordnet eine Service-Level-Vereinbarung einem Prozessticket zu. Das Ticket muss einen Service haben und das SLA muss diesem Service zugeordnet sein. Beispiel:

```
$Self->{'Process::TransitionAction'} = {
  'TA1' => {
    Name => 'Set MySLA SLA',
    Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketSLASet',
    Config => {
      SLA => 'MySLA',
      # or
      SLAID => 123,
    },
  },
};
```

'Name' definiert den Namen der konfigurierten Übergangsaktion.

'SLA' definiert die neue Service-Level-Vereinbarung des Prozesstickets.

'SLAID' definiert die interne ID des neuen SLA.

2.4.5.6.11. TicketStateSet

Ändert den Status eines Prozesstickets. Beispiel:

```
$Self->{'Process::TransitionAction'} = {
  'TA1' => {
    Name => 'Set State to open',
    Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketStateSet',
    Config => {
      State => 'open',
      # or
      StateID => 4,
    },
    PendingTimeDiff => 123,
  },
};
```

'Name' definiert den Namen der konfigurierten Übergangsaktion.

'State' definiert den neuen Status des Prozesstickets.

'StateID' definiert die interne ID des neuen Status.

'PendingTimeDiff' used only for pending type states, defines the time difference in seconds relative (relative to the Transition Action execution time) to set ticket pending time (e.g. 3600 means that the pending time is 1hr after the Transition Action is executed).

2.4.5.6.12. TicketTitleSet

Setzt den Tickettitel eines Prozesstickets. Beispiel:

```
$Self->{'Process::TransitionAction'} = {
  'TA1' => {
    Name => 'Set Ticket Title to Ticket-title',
    Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketTitleSet',
    Config => {
      Title => 'Ticket-title',
    },
  },
};
```

'Name' definiert den Namen der konfigurierten Übergangsaktion.

'Title' definiert den neuen Titel des Tickets.

2.4.5.6.13. TicketTypeSet

Setzt den Tickettyp eines Prozesstickets. Beispiel:

```
$Self->{'Process::TransitionAction'} = {
  'TA1' => {
    Name => 'Set Ticket Type to default',
    Module => 'Kernel::System::ProcessManagement::TransitionAction::TicketTypeSet',
    Config => {
      Type => 'default',
      # or
      # TypeID => '1',
    },
  },
};
```

'Name' definiert den Namen der konfigurierten Übergangsaktion.

'Type' definiert den Namen des Tickettyps.

'TypeID' definiert die interne ID des Tickettyps.

2.4.6. Access Control Lists (ACLs)

Mit der Hilfe von ACLs, können Sie die auswählbaren Werte in Prozesstickets limitieren. Weitere Beschreibungen der ACL-Syntax finden Sie in der ACL-Referenz

2.4.6.1. ACL Konfiguration

ACLs können nur in der Kernel/Config.pm definiert werden. Beispiel:

```
$Self->{TicketAcl}->{'001-ACL-ProcessProperties'} = {
  Properties => {
    Process => {
      ProcessEntityID => ['P1'],
      ActivityEntityID => ['A1'],
      ActivityDialogEntityID => ['AD1'],
    },
  },
};
```

```

}
},
Possible => {
  ActivityDialog => ['AD1', 'AD3'],
},
PossibleNot => {
  ActivityDialog => ['AD3'],
},
};

```

2.4.6.2. 001-ACL-ProzessEigenschaften

Name der ACL-Regel. Für weitere Informationen über ACL-Regeln konsultieren Sie bitte das Kapitel über ACLs.

2.4.6.3. Prozess

This is the section that is used to check if an ACL must be applied. If it has the specified values, the rule is applied. The following values can be used:

2.4.6.3.1. ProcessEntityID

The ID of a process that the process. Matches if the ticket is assigned to this process.

2.4.6.3.2. ActivityEntityID

Die Nummer der Aktivität die dem Prozessticket aktuell zugeordnet ist.

2.4.6.3.3. ActivityDialogEntityID

Die Nummer des Aktivitätendialogs der dem Prozessticket aktuell zugeordnet ist.

2.4.6.4. Possible/PossibleNot Activity Dialog

Hier können Sie eine Liste von Aktivitätsdialogen-Nummern definieren. Diese Liste beschränkt die möglichen Aktivitätsdialoge die dem Agenten in der TicketZoom-Oberfläche angeboten werden.

'Möglich' listet die Aktivitätsdialoge auf, die erlaubt sind. Durch diese Einstellung wird nur 'AD1' und 'AD3' in der Liste der konfigurierten Aktivitätsdialoge angezeigt.

'Nicht Möglich' listet die Aktivitätsdialoge auf, die nicht erlaubt sind. Im oberen Beispiel wird diese Einstellung 'AD3' aus der Liste der konfigurierten Aktivitätsdialoge entfernen.

If both 'Possible' and 'PossibleNot' are specified, the list of configured Activity Dialogs will first be filtered by 'Possible', leaving only 'AD1' and 'AD3' in our example. Then 'Possible-Not' will be applied and filter out 'AD3', so that only 'AD1' remains and is shown as a possible Activity Dialog that the user can use.

Wenn mehrere ACL-Regeln übereinstimmen, wird der Durchschnitt aller passenden Regeln berechnet um die möglichen Aktivitätsdialoge zu bestimmen. Beispiel:

Configured Activity Dialogs: 'AD1', 'AD2', 'AD3', 'AD4', 'AD5', 'AD6', 'AD7'.

```

$self->{TicketAcl}->{'001-ACL-Status'} = {
  Properties => {
    Ticket => {
      Status => 'new',
    }
  },
  Possible => {
    ActivityDialog => ['AD1', 'AD2', 'AD3', 'AD6', 'AD7'],
  },
};

```

```

};
$self->{TicketAcl}->{'002-ACL-Queue'} = {
  Properties => {
    Ticket => {
      Queue => ['Raw']
    }
  },
  Possible => {
    ActivityDialog => ['AD2', 'AD3', 'AD4', 'AD7'],
  },
};
$self->{TicketAcl}->{'003-ACL-Priority'} = {
  Properties => {
    Ticket => {
      Priority => ['3 normal']
    }
  },
  PossibleNot => {
    ActivityDialog => ['AD3', 'AD4'],
  },
};

```

Wenn ein Prozessticket den Zustand "Neu" hat, in der "Raw"-Queue ist und eine Priorität "3 normal" hat, dann werden alle drei ACL-Regeln angewendet werden.

Die erste Regel reduziert die Aktivitätsdialoge von 'AD1', 'AD2', 'AD3', 'AD4', 'AD5', 'AD6', 'AD7' to 'AD1', 'AD2', 'AD3', 'AD6', 'AD7' und verbietet 'AD4' and 'AD5'.

Die zweite Regel reduziert die verbleibende Aktivitätsdialoge weiter. In unserem Beispiel werden 'AD2', 'AD3', 'AD 7' bleiben.

Nun wird die dritte Regel die Liste durch 'PossibleNot' weiter reduzieren. 'AD3' wird von der Liste entfernt. 'AD4' wird nicht entfernt, da er in der Liste nicht an erster Stelle steht. Am Ende bleiben 'AD2' und 'AD 7' als mögliche Aktivitätsdialoge, die der Benutzer verwenden kann.

It is also possible to limit the processes that can be displayed in the "New process ticket" screen. The functionality is similar to limiting the Activity Dialogs with one exception: The ACLs could only be based on Users.

Siehe Beispiele:

```

$self->{TicketAcl}->{'200-ACL-Process'} = {
  # match properties
  Properties => {
    User => {
      UserID => [2, 3],
    },
  },
  Possible => {
    Process => ['P1', 'P2', 'P3'],
  },
  PossibleNot => {
    Process => ['P4'],
  },
};

```

```

$self->{TicketAcl}->{'201-ACL-Process'} = {
  # match properties
  Properties => {
    User => {
      Group_rw => [ 'MyGroup' ],
    },
  },
  Possible => {

```



```
    Process => ['P1', 'P2', 'P3'],
  },
  PossibleNot => {
    Process => ['P4'],
  },
};
```

```
$Self->{TicketAcl}->{'202-ACL-Process'} = {
  # match properties
  Properties => {
    User => {
      Role => [ 'MyRole' ],
    },
  },
  Possible => {
    Process => ['P1', 'P2', 'P3'],
  },
  PossibleNot => {
    Process => ['P4'],
  },
};
```

3. Localization of the OTRS Front End

Procedures for localization for the OTRS framework, steps to be followed to create a new language translation, as well as procedures for translation customizations, can be found in the "[Language Translations](http://otrs.github.io/doc)" chapter from the developer manual on <http://otrs.github.io/doc>.

Kapitel 6.

Leistungsverbesserung

Presented below is a list of performance enhancing techniques for your OTRS installation, including configuration, coding, memory use, and more.

1. OTRS

Es gibt verschiedene Ansätze zur Leistungssteigerung von OTRS.

1.1. TicketIndexModule

There are two backend modules for the index for the ticket queue view:

- Using `Kernel::System::Ticket::IndexAccelerator::RuntimeDB` (default), generate each queue view on the fly from the ticket table. You will not have performance trouble until you have about 60,000 open tickets in your system.
- `Kernel::System::Ticket::IndexAccelerator::StaticDB`, the most powerful module, should be used when you have above 80,000 open tickets. It uses an extra `ticket_index` table, which works like a view. Use `bin/otrs.Console.pl Maint::Ticket::QueueIndexRebuild` for generating an initial index after switching backends.

Sie können den `IndexAccelerator` per `SysConfig` einstellen.

1.2. TicketStorageModule

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

- Configure `Kernel::System::Ticket::ArticleStorageDB` (default) to store attachments, etc., in the database. Note: Don't use it with large setups.

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).

- Configure `Kernel::System::Ticket::ArticleStorageFS` to store attachments etc. on the local file system. Note: Recommended for large setups.

Pro: Schneller!

Con: Your web server user should be the 'otrs' user. Also, if you have multiple front-end servers, you should make sure the filesystem is shared between the servers. Place it on an NFS share or preferably a SAN or similar solution.

Note: you can switch from one back-end to the other on the fly. You can switch the backend in the `SysConfig`, and then run the command line utility `bin/otrs.Console.pl Admin::Article::StorageSwitch` to put the articles from the database onto the filesystem or the other way around. You can use the `--target` option to specify the target backend. Please note that the entire process can take considerable time to run, depending on the number of articles you have and the available CPU power and/or network capacity.

```
shell> bin/otrs.Console.pl Admin::Article::StorageSwitch --target ArticleStorageFS
```

Script: Switching storage back-ends from database to filesystem.

If you want to keep old attachments in the database, you can activate the SysConfig option `Ticket::StorageModule::CheckAllBackends` to make sure OTRS will still find them.

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.

Tickets that match certain criteria can be marked as "archived". These tickets are not accessed if you do a regular ticket search or run a Generic Agent job. The system itself does not have to deal with a huge amount of tickets any longer as only the "latest" tickets are taken into consideration when using OTRS. This can result in a huge performance gain on large systems.

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

On the Admin page, select `GenericAgent` and add a new job there.

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.

1.4. Cache

OTRS caches a lot of temporary data in `/opt/otrs/var/tmp`. Please make sure that this uses a high performance file system/storage. If you have enough RAM, you can also try to put this directory on a ramdisk like this:

```
shell> /opt/otrs/bin/otrs.Console.pl Maint::Session::DeleteAll
shell> /opt/otrs/bin/otrs.Console.pl Maint::Cache::Delete
shell> sudo mount -o size=16G -t tmpfs none /opt/otrs/var/tmp

# add persistent mount point in /etc/fstab
```

Anmerkung

Bitte beachten Sie, dass dies ein flüchtiger Speicher ist, welcher nach einem Server Neustart verloren ist. Alle Sitzungen (wenn Sie diese im Filesystem speichern) und die Cache Daten sind verloren.

There is also a centralized memcached based Cache backend available for purchase from OTRS Group.

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 (Tabellen) 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;
```

Skript: Optimierung von Datenbanktabellen.

2.2. PostgreSQL

PostgreSQL is best tuned by modifying the `postgresql.conf` file in your PostgreSQL data directory. For advice on how to do this, reference the following articles:

- <http://www.revsys.com/writings/postgresql-performance.html>
- <http://varlena.com/GeneralBits/Tidbits/perf.html>
- http://varlena.com/GeneralBits/Tidbits/annotated_conf_e.html

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

Bei wirklich großen Installationen (über 1000 neue Tickets am Tag, über 40 Agenten) ist es eine sehr gute Idee, den Artikel "Choosing the Right Strategy" (in englisch) zu lesen (<http://perl.apache.org/docs/1.0/guide/strategy.html>).

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.

Anhang A. Weitere Quellen

otrs.com/?lang=de

The OTRS website with source code, documentation and news is available at www.otrs.com. Here you can also find information about professional services and OTRS Administrator training seminars from OTRS Group, the creator of OTRS.

Mailinglisten

Tabelle A.1. Mailinglisten

Name & URL	Beschreibung
announce@otrs.org	Ankündigungen zu OTRS (neue Versionen, Updates, ...) mit wenig Verkehr.
otrs@otrs.org	Englische Benutzer-Fragen und -Diskussionen rund um OTRS mit wenigen bis vielen Anfragen.
otrs-de@otrs.org	Deutsche Benutzer-Fragen und -Diskussionen rund um OTRS mit wenigen bis vielen Anfragen.
dev@otrs.org	Mailingliste für die Entwickler von OTRS mit wenig Verkehr.

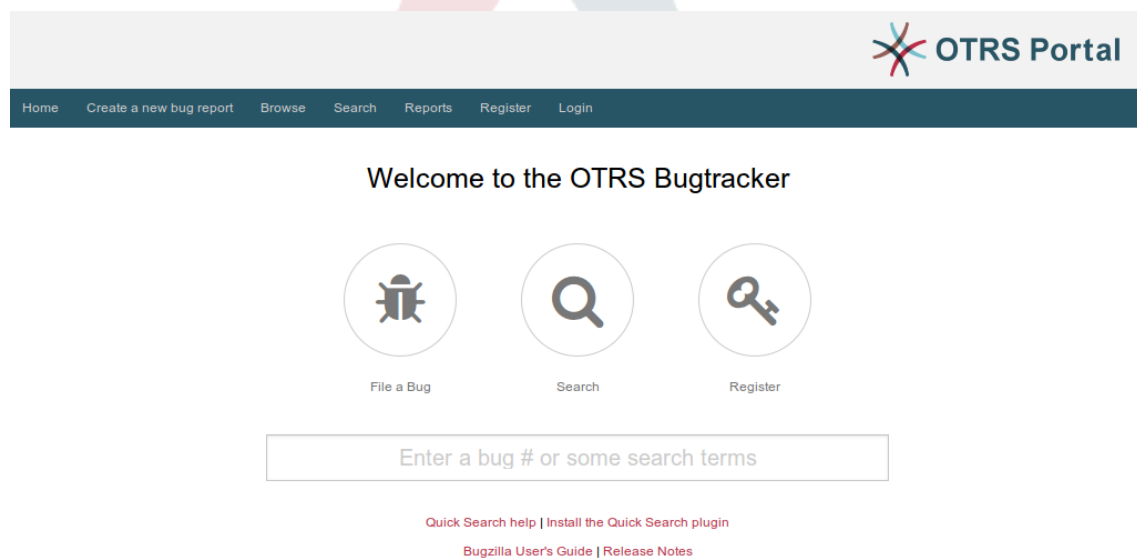
Übersetzungen

Sie können unter [Transifex](#) dabei helfen OTRS in Ihre Sprache zu übersetzen.

Fehler berichten

To report software defects, please visit <http://bugs.otrs.org/> (see figure below). Please take note of the difference between a bug and a configuration issue. Configuration issues are problems that you encounter when setting a system, or general questions regarding the use of OTRS. Bug reports should only be used for issues with the source code of OTRS or other open source OTRS modules itself. For configuration issues, you should either use the [commercial support, available from OTRS](#), or the public mailing lists.

Abbildung A.1. Fehlerverfolgungssystem



Anhang B. Referenz der Konfigurationseinstellungen

1. CloudService

CloudService → CloudService::Admin::ModuleRegistration

CloudService::Admin::Module###100-SupportDataCollector

Cloud service admin module registration for the transport layer.

Standardwert:

```
$Self->{'CloudService::Admin::Module'}->{'100-SupportDataCollector'} = {
  'ConfigDialog' => 'AdminCloudServiceSupportDataCollector',
  'Description' => 'Configure sending of support data to OTRS Group for improved support.',
  'Icon' => 'fa fa-compass',
  'Name' => 'Support data collector'
};
```

CloudService::Admin::Module###200-SMS

Cloud service admin module registration for the transport layer.

Standardwert:

```
$Self->{'CloudService::Admin::Module'}->{'200-SMS'} = {
  'ConfigDialog' => 'AdminCloudServiceSMS',
  'Description' => 'This will allow the system to send text messages via SMS.',
  'Icon' => 'fa fa-mobile',
  'IsOTRSBusiness' => '1',
  'Name' => 'SMS'
};
```

CloudService → Core

CloudServices::Disabled

Disables the communication between this system and OTRS Group servers that provides cloud services. If active, some functionality will be lost such as system registration, support data sending, upgrading to and use of OTRS Business Solution™, OTRS Verify™, OTRS News and product News dashboard widgets, among others.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'CloudServices::Disabled'} = '0';
```

CloudService → Frontend::Agent::ModuleNotify

Frontend::NotifyModule###1000-CloudServicesDisabled

Defines the module to display a notification if cloud services are disabled.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::NotifyModule'}->{'1000-CloudServicesDisabled'} = {
  'Group' => 'admin',
  'Module' => 'Kernel::Output::HTML::Notification::AgentCloudServicesDisabled'
};
```

2. Daemon

Daemon → Core::Daemon::ModuleRegistration

DaemonModules###SchedulerGenericAgentTaskManager

The daemon registration for the scheduler generic agent task manager.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'DaemonModules'}->{'SchedulerGenericAgentTaskManager'} = {  
  'Module' => 'Kernel::System::Daemon::DaemonModules::SchedulerGenericAgentTaskManager'  
};
```

DaemonModules###SchedulerCronTaskManager

The daemon registration for the scheduler cron task manager.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'DaemonModules'}->{'SchedulerCronTaskManager'} = {  
  'Module' => 'Kernel::System::Daemon::DaemonModules::SchedulerCronTaskManager'  
};
```

DaemonModules###SchedulerFutureTaskManager

The daemon registration for the scheduler future task manager.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'DaemonModules'}->{'SchedulerFutureTaskManager'} = {  
  'Module' => 'Kernel::System::Daemon::DaemonModules::SchedulerFutureTaskManager'  
};
```

DaemonModules###SchedulerTaskWorker

The daemon registration for the scheduler task worker.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'DaemonModules'}->{'SchedulerTaskWorker'} = {  
  'Module' => 'Kernel::System::Daemon::DaemonModules::SchedulerTaskWorker'  
};
```

Daemon → Core::Log

Daemon::Log::DaysToKeep

Defines the number of days to keep the daemon log files.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::Log::DaysToKeep'} = '1';
```

Daemon::Log::STDOUT

If enabled the daemon will redirect the standard output stream to a log file.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:


```
$Self->{'Daemon::Log::STDOUT'} = '0';
```

Daemon::Log::STDERR

If enabled the daemon will redirect the standard error stream to a log file.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Daemon::Log::STDERR'} = '1';
```

Daemon → Core::Web

Loader::Agent::CommonCSS###001-Daemon

List of CSS files to always be loaded for the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Loader::Agent::CommonCSS'}->{'001-Daemon'} = [
  'Core.Agent.Daemon.css'
];
```

Loader::Agent::CommonJS###001-Daemon

List of JS files to always be loaded for the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Loader::Agent::CommonJS'}->{'001-Daemon'} = [
  'Core.Agent.Daemon.js'
];
```

Daemon → Daemon::SchedulerCronTaskManager::Task

Daemon::SchedulerCronTaskManager::Task###CoreCacheCleanup

Delete expired cache from core modules.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'CoreCacheCleanup'} = {
  'Function' => 'Cleanup',
  'MaximumParallelInstances' => '1',
  'Module' => 'Kernel::System::Cache',
  'Params' => [
    'Expired',
    '1'
  ],
  'Schedule' => '20 0 * * 0',
  'TaskName' => 'CoreCacheCleanup'
};
```

Daemon::SchedulerCronTaskManager::Task###WebUploadCacheCleanup

Delete expired upload cache hourly.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'WebUploadCacheCleanup'} = {
  'Function' => 'FormIDCleanup',
  'MaximumParallelInstances' => '1',
  'Module' => 'Kernel::System::Web::UploadCache',
  'Params' => [],
  'Schedule' => '46 * * * *',
  'TaskName' => 'WebUploadCacheCleanup'
};
```

Daemon::SchedulerCronTaskManager::Task###LoaderCacheDelete

Delete expired loader cache weekly (Sunday mornings).

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'LoaderCacheDelete'} = {
  'Function' => 'CacheDelete',
  'MaximumParallelInstances' => '1',
  'Module' => 'Kernel::System::Loader',
  'Params' => [],
  'Schedule' => '30 0 * * 0',
  'TaskName' => 'LoaderCacheDelete'
};
```

Daemon::SchedulerCronTaskManager::Task###FetchMail

Fetch emails via fetchmail.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'FetchMail'} = {
  'Function' => 'Fetch',
  'MaximumParallelInstances' => '1',
  'Module' => 'Kernel::System::FetchMail',
  'Params' => [],
  'Schedule' => '* /5 * * * *',
  'TaskName' => 'FetchMail'
};
```

Daemon::SchedulerCronTaskManager::Task###FetchMailSSL

Fetch emails via fetchmail (using SSL).

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'FetchMailSSL'} = {
  'Function' => 'Fetch',
  'MaximumParallelInstances' => '1',
  'Module' => 'Kernel::System::FetchMail',
  'Params' => [
    'SSL',
    '1'
  ],
  'Schedule' => '* /5 * * * *',
  'TaskName' => 'FetchMailSSL'
};
```

Daemon::SchedulerCronTaskManager::Task###GenerateDashboardStats

Generate dashboard statistics.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'GenerateDashboardStats'} = {
  'Function' => 'Execute',
  'MaximumParallelInstances' => '1',
  'Module' => 'Kernel::System::Console::Command::Maint::Stats::Dashboard::Generate',
  'Params' => [],
  'Schedule' => '5 * * * *',
  'TaskName' => 'GenerateDashboardStats'
};
```

Daemon::SchedulerCronTaskManager::Task###EscalationCheck

Triggers ticket escalation events and notification events for escalation.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'EscalationCheck'} = {
  'Function' => 'Execute',
```

```
'MaximumParallelInstances' => '1',
'Module' => 'Kernel::System::Console::Command::Maint::Ticket::EscalationCheck',
'Params' => [],
'Schedule' => '*/*5 * * * *',
'TaskName' => 'EscalationCheck'
};
```

Daemon::SchedulerCronTaskManager::Task###TicketPendingCheck

Process pending tickets.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'TicketPendingCheck'} = {
'Function' => 'Execute',
'MaximumParallelInstances' => '1',
'Module' => 'Kernel::System::Console::Command::Maint::Ticket::PendingCheck',
'Params' => [],
'Schedule' => '45 */2 * * *',
'TaskName' => 'TicketPendingCheck'
};
```

Daemon::SchedulerCronTaskManager::Task###SpoolMailsReprocess

Reprocess mails from spool directory that could not be imported in the first place.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'SpoolMailsReprocess'} = {
'Function' => 'Execute',
'MaximumParallelInstances' => '1',
'Module' =>
'Kernel::System::Console::Command::Maint::PostMaster::SpoolMailsReprocess',
'Params' => [],
'Schedule' => '10 0 * * *',
'TaskName' => 'SpoolMailsReprocess'
};
```

Daemon::SchedulerCronTaskManager::Task###MailAccountFetch

Fetch incoming emails from configured mail accounts.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'MailAccountFetch'} = {
'Function' => 'Execute',
'MaximumParallelInstances' => '1',
'Module' => 'Kernel::System::Console::Command::Maint::PostMaster::MailAccountFetch',
'Params' => [],
'Schedule' => '*/*10 * * * *',
'TaskName' => 'MailAccountFetch'
};
```

Daemon::SchedulerCronTaskManager::Task###TicketAcceleratorRebuild

Rebuild the ticket index for AgentTicketQueue.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'TicketAcceleratorRebuild'} = {
'Function' => 'TicketAcceleratorRebuild',
'MaximumParallelInstances' => '1',
'Module' => 'Kernel::System::Ticket',
'Params' => [],
'Schedule' => '01 01 * * *',
'TaskName' => 'TicketAcceleratorRebuild'
};
```

Daemon::SchedulerCronTaskManager::Task###SessionDeleteExpired

Delete expired sessions.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'SessionDeleteExpired'} = {
```

```
'Function' => 'Execute',
'MaximumParallelInstances' => '1',
'Module' => 'Kernel::System::Console::Command::Maint::Session::DeleteExpired',
'Params' => [],
'Schedule' => '55 */2 * * *',
'TaskName' => 'SessionDeleteExpired'
};
```

Daemon::SchedulerCronTaskManager::Task###TicketUnlockTimeout

Unlock tickets that are past their unlock timeout.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'TicketUnlockTimeout'} = {
'Function' => 'Execute',
'MaximumParallelInstances' => '1',
'Module' => 'Kernel::System::Console::Command::Maint::Ticket::UnlockTimeout',
'Params' => [],
'Schedule' => '35 * * * *',
'TaskName' => 'TicketUnlockTimeout'
};
```

Daemon::SchedulerCronTaskManager::Task###RenewCustomerSMIMECertificates

Renew existing SMIME certificates from customer backend. Note: SMIME and SMIME::FetchFromCustomer needs to be active in SysConfig and customer backend needs to be configured to fetch UserSMIMECertificate attribute.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'RenewCustomerSMIMECertificates'} =
{
'Function' => 'Execute',
'MaximumParallelInstances' => '1',
'Module' =>
'Kernel::System::Console::Command::Maint::SMIME::CustomerCertificate::Renew',
'Params' => [],
'Schedule' => '02 02 * * *',
'TaskName' => 'RenewCustomerSMIMECertificates'
};
```

Daemon::SchedulerCronTaskManager::Task###Custom1

Executes a custom command or module. Note: if module is used, function is required.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'Custom1'} = {
'Function' => '',
'MaximumParallelInstances' => '1',
'Module' => '',
'Params' => [],
'Schedule' => '* * * * *',
'TaskName' => 'Custom1'
};
```

Daemon::SchedulerCronTaskManager::Task###Custom2

Executes a custom command or module. Note: if module is used, function is required.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'Custom2'} = {
'Function' => '',
'MaximumParallelInstances' => '1',
'Module' => '',
'Params' => [],
'Schedule' => '* * * * *',
};
```

```
'TaskName' => 'Custom2'  
};
```

Daemon::SchedulerCronTaskManager::Task###Custom3

Executes a custom command or module. Note: if module is used, function is required.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'Custom3'} = {  
  'Function' => '',  
  'MaximumParallelInstances' => '1',  
  'Module' => '',  
  'Params' => [],  
  'Schedule' => '* * * * *',  
  'TaskName' => 'Custom3'  
};
```

Daemon::SchedulerCronTaskManager::Task###Custom4

Executes a custom command or module. Note: if module is used, function is required.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'Custom4'} = {  
  'Function' => '',  
  'MaximumParallelInstances' => '1',  
  'Module' => '',  
  'Params' => [],  
  'Schedule' => '* * * * *',  
  'TaskName' => 'Custom4'  
};
```

Daemon::SchedulerCronTaskManager::Task###Custom5

Executes a custom command or module. Note: if module is used, function is required.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'Custom5'} = {  
  'Function' => '',  
  'MaximumParallelInstances' => '1',  
  'Module' => '',  
  'Params' => [],  
  'Schedule' => '* * * * *',  
  'TaskName' => 'Custom5'  
};
```

Daemon::SchedulerCronTaskManager::Task###Custom6

Executes a custom command or module. Note: if module is used, function is required.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'Custom6'} = {  
  'Function' => '',  
  'MaximumParallelInstances' => '1',  
  'Module' => '',  
  'Params' => [],  
  'Schedule' => '* * * * *',  
  'TaskName' => 'Custom6'  
};
```

Daemon::SchedulerCronTaskManager::Task###Custom7

Executes a custom command or module. Note: if module is used, function is required.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'Custom7'} = {
  'Function' => '',
  'MaximumParallelInstances' => '1',
  'Module' => '',
  'Params' => [],
  'Schedule' => '* * * * *',
  'TaskName' => 'Custom7'
};
```

Daemon::SchedulerCronTaskManager::Task###Custom8

Executes a custom command or module. Note: if module is used, function is required.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'Custom8'} = {
  'Function' => '',
  'MaximumParallelInstances' => '1',
  'Module' => '',
  'Params' => [],
  'Schedule' => '* * * * *',
  'TaskName' => 'Custom8'
};
```

Daemon::SchedulerCronTaskManager::Task###Custom9

Executes a custom command or module. Note: if module is used, function is required.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'Custom9'} = {
  'Function' => '',
  'MaximumParallelInstances' => '1',
  'Module' => '',
  'Params' => [],
  'Schedule' => '* * * * *',
  'TaskName' => 'Custom9'
};
```

Daemon::SchedulerCronTaskManager::Task###GenericAgentFile1

Run file based generic agent jobs (Note: module name need needs to be specified in -configuration-module param e.g. "Kernel::System::GenericAgent").

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'GenericAgentFile1'} = {
  'Function' => 'Execute',
  'MaximumParallelInstances' => '1',
  'Module' => 'Kernel::System::Console::Command::Maint::GenericAgent::Run',
  'Params' => [
    '--configuration-module',
    '<ModuleName>'
  ],
  'Schedule' => '* / 20 * * * *',
  'TaskName' => 'GenericAgentFile1'
};
```

Daemon::SchedulerCronTaskManager::Task###GenericAgentFile2

Run file based generic agent jobs (Note: module name need needs to be specified in -configuration-module param e.g. "Kernel::System::GenericAgent").

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'GenericAgentFile2'} = {
  'Function' => 'Execute',
  'MaximumParallelInstances' => '1',
  'Module' => 'Kernel::System::Console::Command::Maint::GenericAgent::Run',
  'Params' => [
    '--configuration-module',
    '<ModuleName>'
  ],
  'Schedule' => '*/*20 * * * *',
  'TaskName' => 'GenericAgentFile2'
};
```

Daemon::SchedulerCronTaskManager::Task###GenericAgentFile3

Run file based generic agent jobs (Note: module name need needs to be specified in -configuration-module param e.g. "Kernel::System::GenericAgent").

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'GenericAgentFile3'} = {
  'Function' => 'Execute',
  'MaximumParallelInstances' => '1',
  'Module' => 'Kernel::System::Console::Command::Maint::GenericAgent::Run',
  'Params' => [
    '--configuration-module',
    '<ModuleName>'
  ],
  'Schedule' => '*/*20 * * * *',
  'TaskName' => 'GenericAgentFile3'
};
```

Daemon::SchedulerCronTaskManager::Task###GenericAgentFile4

Run file based generic agent jobs (Note: module name need needs to be specified in -configuration-module param e.g. "Kernel::System::GenericAgent").

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'GenericAgentFile4'} = {
  'Function' => 'Execute',
  'MaximumParallelInstances' => '1',
  'Module' => 'Kernel::System::Console::Command::Maint::GenericAgent::Run',
  'Params' => [
    '--configuration-module',
    '<ModuleName>'
  ],
  'Schedule' => '*/*20 * * * *',
  'TaskName' => 'GenericAgentFile4'
};
```

Daemon::SchedulerCronTaskManager::Task###GenericAgentFile5

Run file based generic agent jobs (Note: module name need needs to be specified in -configuration-module param e.g. "Kernel::System::GenericAgent").

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'GenericAgentFile5'} = {
  'Function' => 'Execute',
  'MaximumParallelInstances' => '1',
  'Module' => 'Kernel::System::Console::Command::Maint::GenericAgent::Run',
  'Params' => [
```

```
'--configuration-module',
'<ModuleName>'
],
'Schedule' => '*/*20 * * * *',
'TaskName' => 'GenericAgentFile5'
};
```

Daemon::SchedulerCronTaskManager::Task###RegistrationUpdateSend

Sends registration information to OTRS group.

Diese Einstellung kann nicht geändert werden.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'RegistrationUpdateSend'} = {
'Function' => 'Execute',
'MaximumParallelInstances' => '1',
'Module' => 'Kernel::System::Console::Command::Maint::Registration::UpdateSend',
'Params' => [],
'Schedule' => '30 * * * *',
'TaskName' => 'RegistrationUpdateSend'
};
```

Daemon::SchedulerCronTaskManager::Task###SupportDataCollectAsynchronous

Collect support data for asynchronous plug-in modules.

Diese Einstellung kann nicht geändert werden.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'SupportDataCollectAsynchronous'} =
{
'Function' => 'Execute',
'MaximumParallelInstances' => '1',
'Module' =>
'Kernel::System::Console::Command::Maint::SupportData::CollectAsynchronous',
'Params' => [],
'Schedule' => '1 * * * *',
'TaskName' => 'SupportDataCollectAsynchronous'
};
```

Daemon::SchedulerCronTaskManager::Task###OTRSBusinessEntitlementCheck

Checks the entitlement status of OTRS Business Solution™.

Diese Einstellung kann nicht geändert werden.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'OTRSBusinessEntitlementCheck'} =
{
'Function' => 'Execute',
'MaximumParallelInstances' => '1',
'Module' => 'Kernel::System::Console::Command::Maint::OTRSBusiness::EntitlementCheck',
'Params' => [],
'Schedule' => '25,45 */1 * * *',
'TaskName' => 'OTRSBusinessEntitlementCheck'
};
```

Daemon::SchedulerCronTaskManager::Task###OTRSBusinessAvailabilityCheck

Checks the availability of OTRS Business Solution™ for this system.

Diese Einstellung kann nicht geändert werden.

Standardwert:

```
$Self->{'Daemon::SchedulerCronTaskManager::Task'}->{'OTRSBusinessAvailabilityCheck'} =
{
'Function' => 'Execute',
'MaximumParallelInstances' => '1',
```



```
'Module' =>
'Kernel::System::Console::Command::Maint::OTRSBusiness::AvailabilityCheck',
'Params' => [],
'Schedule' => '15,35,55 */1 * * *',
'TaskName' => 'OTRSBusinessAvailabilityCheck'
};
```

Daemon → Daemon::SchedulerGenericAgentTaskManager

Daemon::SchedulerGenericAgentTaskManager::TicketLimit

Defines the maximum number of affected tickets per job.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Daemon::SchedulerGenericAgentTaskManager::TicketLimit'} = '4000';
```

Daemon::SchedulerGenericAgentTaskManager::SleepTime

Defines a sleep time in microseconds between tickets while they are been processed by a job.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Daemon::SchedulerGenericAgentTaskManager::SleepTime'} = '0';
```

Daemon → Daemon::SchedulerGenericInterfaceTaskManager

Daemon::SchedulerGenericInterfaceTaskManager::FutureTaskTimeDiff

Defines the default the number of seconds (from current time) to re-schedule a generic interface failed task.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Daemon::SchedulerGenericInterfaceTaskManager::FutureTaskTimeDiff'} = '300';
```

Daemon → Daemon::SchedulerTaskWorker

Daemon::SchedulerTaskWorker::MaximumWorkers

Defines the maximum number of tasks to be executed as the same time.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Daemon::SchedulerTaskWorker::MaximumWorkers'} = '5';
```

Daemon::SchedulerTaskWorker::NotificationRecipientEmail

Specifies the email addresses to get notification messages from scheduler tasks.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Daemon::SchedulerTaskWorker::NotificationRecipientEmail'} = 'root@localhost';
```

Daemon → Frontend::Admin::ModuleRegistration

Frontend::Module###AgentDaemonInfo

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentDaemonInfo'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Title' => 'Shows information on how to start OTRS Daemon'
};
```

Daemon → Frontend::Agent::ModuleNotify

Frontend::NotifyModule###8000-Daemon-Check

Defines the module to display a notification in the agent interface if the OTRS Daemon is not running.

Standardwert:

```
$Self->{'Frontend::NotifyModule'}->{'8000-Daemon-Check'} = {
  'Module' => 'Kernel::Output::HTML::Notification::DaemonCheck'
};
```

3. DynamicFields

DynamicFields → DynamicFields::Driver::Registration

DynamicFields::Driver###Text

DynamicField backend registration.

Standardwert:

```
$Self->{'DynamicFields::Driver'}->{'Text'} = {
  'ConfigDialog' => 'AdminDynamicFieldText',
  'DisplayName' => 'Text',
  'Module' => 'Kernel::System::DynamicField::Driver::Text'
};
```

DynamicFields::Driver###TextArea

DynamicField backend registration.

Standardwert:

```
$Self->{'DynamicFields::Driver'}->{'TextArea'} = {
  'ConfigDialog' => 'AdminDynamicFieldText',
  'DisplayName' => 'Textarea',
  'Module' => 'Kernel::System::DynamicField::Driver::TextArea'
};
```

DynamicFields::Driver###Checkbox

DynamicField backend registration.

Standardwert:

```
$Self->{'DynamicFields::Driver'}->{'Checkbox'} = {
  'ConfigDialog' => 'AdminDynamicFieldCheckbox',
  'DisplayName' => 'Checkbox',
  'Module' => 'Kernel::System::DynamicField::Driver::Checkbox'
};
```

DynamicFields::Driver###Dropdown

DynamicField backend registration.

Standardwert:

```
$Self->{'DynamicFields::Driver'}->{'Dropdown'} = {
  'ConfigDialog' => 'AdminDynamicFieldDropdown',
};
```

```
'DisplayName' => 'Dropdown',
'Module' => 'Kernel::System::DynamicField::Driver::Dropdown'
};
```

DynamicFields::Driver###DateTime

DynamicField backend registration.

Standardwert:

```
$Self->{'DynamicFields::Driver'}->{'DateTime'} = {
  'ConfigDialog' => 'AdminDynamicFieldDateTime',
  'DisplayName' => 'Date / Time',
  'Module' => 'Kernel::System::DynamicField::Driver::DateTime'
};
```

DynamicFields::Driver###Date

DynamicField backend registration.

Standardwert:

```
$Self->{'DynamicFields::Driver'}->{'Date'} = {
  'ConfigDialog' => 'AdminDynamicFieldDateTime',
  'DisplayName' => 'Date',
  'Module' => 'Kernel::System::DynamicField::Driver::Date'
};
```

DynamicFields::Driver###Multiselect

DynamicField backend registration.

Standardwert:

```
$Self->{'DynamicFields::Driver'}->{'Multiselect'} = {
  'ConfigDialog' => 'AdminDynamicFieldMultiselect',
  'DisplayName' => 'Multiselect',
  'ItemSeparator' => ', ',
  'Module' => 'Kernel::System::DynamicField::Driver::Multiselect'
};
```

DynamicFields → DynamicFields::ObjectType::Registration

DynamicFields::ObjectType###CustomerCompany

DynamicField object registration.

Standardwert:

```
$Self->{'DynamicFields::ObjectType'}->{'CustomerCompany'} = {
  'DisplayName' => 'Customer',
  'Module' => 'Kernel::System::DynamicField::ObjectType::CustomerCompany',
  'Prio' => '120',
  'UseObjectName' => '1'
};
```

DynamicFields::ObjectType###CustomerUser

DynamicField object registration.

Standardwert:

```
$Self->{'DynamicFields::ObjectType'}->{'CustomerUser'} = {
  'DisplayName' => 'Customer User',
  'Module' => 'Kernel::System::DynamicField::ObjectType::CustomerUser',
  'Prio' => '130',
  'UseObjectName' => '1'
};
```

DynamicFields::ObjectType###Article

DynamicField object registration.

Standardwert:

```
$Self->{'DynamicFields::ObjectType'}->{'Article'} = {
  'DisplayName' => 'Article',
  'Module' => 'Kernel::System::DynamicField::ObjectType::Article',
  'Prio' => '110'
};
```

DynamicFields::ObjectType###Ticket

DynamicField object registration.

Standardwert:

```
$Self->{'DynamicFields::ObjectType'}->{'Ticket'} = {
  'DisplayName' => 'Ticket',
  'Module' => 'Kernel::System::DynamicField::ObjectType::Ticket',
  'Prio' => '100'
};
```

DynamicFields → Frontend::Admin::ModuleRegistration

Frontend::Module###AdminDynamicField

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminDynamicField'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.DynamicField.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.DynamicField.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Create and manage dynamic fields.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Dynamic Fields',
    'Prio' => '1000'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Dynamic Fields GUI'
};
```

Frontend::Module###AdminDynamicFieldText

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminDynamicFieldText'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.DynamicField.js',
      'Core.Agent.Admin.DynamicFieldText.js'
    ]
  },
  'Title' => 'Dynamic Fields Text Backend GUI'
};
```

Frontend::Module###AdminDynamicFieldCheckbox

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminDynamicFieldCheckbox'} = {  
  'Description' => 'This module is part of the admin area of OTRS.',  
  'Group' => [  
    'admin'  
  ],  
  'Loader' => {  
    'JavaScript' => [  
      'Core.Agent.Admin.DynamicField.js',  
      'Core.Agent.Admin.DynamicFieldCheckbox.js'  
    ]  
  },  
  'Title' => 'Dynamic Fields Checkbox Backend GUI'  
};
```

Frontend::Module###AdminDynamicFieldDropdown

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminDynamicFieldDropdown'} = {  
  'Description' => 'This module is part of the admin area of OTRS.',  
  'Group' => [  
    'admin'  
  ],  
  'Loader' => {  
    'CSS' => [  
      'Core.Agent.Admin.DynamicField.css'  
    ],  
    'JavaScript' => [  
      'Core.Agent.Admin.DynamicField.js',  
      'Core.Agent.Admin.DynamicFieldDropdown.js'  
    ]  
  },  
  'Title' => 'Dynamic Fields Drop-down Backend GUI'  
};
```

Frontend::Module###AdminDynamicFieldDateTime

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminDynamicFieldDateTime'} = {  
  'Description' => 'This module is part of the admin area of OTRS.',  
  'Group' => [  
    'admin'  
  ],  
  'Loader' => {  
    'CSS' => [  
      'Core.Agent.Admin.DynamicField.css'  
    ],  
    'JavaScript' => [  
      'Core.Agent.Admin.DynamicField.js',  
      'Core.Agent.Admin.DynamicFieldDateTime.js'  
    ]  
  },  
  'Title' => 'Dynamic Fields Date Time Backend GUI'  
};
```

Frontend::Module###AdminDynamicFieldMultiselect

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminDynamicFieldMultiselect'} = {  
  'Description' => 'This module is part of the admin area of OTRS.',  
  'Group' => [  
    'admin'  
  ],  
};
```

```
'Loader' => {
  'CSS' => [
    'Core.Agent.Admin.DynamicField.css'
  ],
  'JavaScript' => [
    'Core.Agent.Admin.DynamicField.js',
    'Core.Agent.Admin.DynamicFieldMultiselect.js'
  ]
},
'Title' => 'Dynamic Fields Multiselect Backend GUI'
};
```

DynamicFields → Frontend::Agent::Preferences

PreferencesGroups###DynamicField

Defines the config parameters of this item, to be shown in the preferences view.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'DynamicField'} = {
  'Active' => '1',
  'Block' => 'Input',
  'Column' => 'Other Settings',
  'Data' => "[% Env('UserDynamicField_NameX') %]",
  'Key' => 'Default value for NameX',
  'Label' => 'NameX',
  'Module' => 'Kernel::Output::HTML::Preferences::Generic',
  'PrefKey' => 'UserDynamicField_NameX',
  'Prio' => '7000'
};
```

PreferencesGroups###DynamicFieldsOverviewPageShown

Parameters for the pages (in which the dynamic fields are shown) of the dynamic fields overview.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'DynamicFieldsOverviewPageShown'} = {
  'Active' => '0',
  'Column' => 'Other Settings',
  'Data' => {
    '10' => '10',
    '15' => '15',
    '20' => '20',
    '25' => '25',
    '30' => '30',
    '35' => '35'
  },
  'DataSelected' => '25',
  'Key' => 'Dynamic fields limit per page for Dynamic Fields Overview',
  'Label' => 'Dynamic Fields Overview Limit',
  'Module' => 'Kernel::Output::HTML::Preferences::Generic',
  'PrefKey' => 'AdminDynamicFieldsOverviewPageShown',
  'Prio' => '8000'
};
```

4. Framework

Framework → Core

SecureMode

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SecureMode'} = '0';
```

Frontend::DebugMode

Enables or disables the debug mode over frontend interface.

Standardwert:

```
$Self->{'Frontend::DebugMode'} = '0';
```

Frontend::AjaxDebug

Delivers extended debugging information in the frontend in case any AJAX errors occur, if enabled.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::AjaxDebug'} = '0';
```

Frontend::TemplateCache

Enables or disables the caching for templates. **WARNING: Do NOT disable template caching for production environments for it will cause a massive performance drop! This setting should only be disabled for debugging reasons!**

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::TemplateCache'} = '1';
```

ConfigLevel

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'ConfigLevel'} = '100';
```

ConfigImportAllowed

Controls if the admin is allowed to import a saved system configuration in SysConfig.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'ConfigImportAllowed'} = '1';
```

ProductName

Defines the name of the application, shown in the web interface, tabs and title bar of the web browser.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'ProductName'} = 'OTRS 6';
```

SystemID

Defines the system identifier. Every ticket number and http session string contains this ID. This ensures that only tickets which belong to your system will be processed as follow-ups (useful when communicating between two instances of OTRS).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SystemID'} = '10';
```

FQDN

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

SupportDataCollector::HTTPHostname

Defines the HTTP hostname for the support data collection with the public module 'PublicSupportDataCollector' (e.g. used from the OTRS Daemon).

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'SupportDataCollector::HTTPHostname'} = '';
```

NodeID

Defines the cluster node identifier. This is only used in cluster configurations where there is more than one OTRS frontend system. Note: only values from 1 to 99 are allowed.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'NodeID'} = '1';
```

HttpType

Defines the type of protocol, used by the web server, to serve the application. If https protocol will be used instead of plain http, it must be specified here. Since this has no affect on the web server's settings or behavior, it will not change the method of access to the application and, if it is wrong, it will not prevent you from logging into the application. This setting is only used as a variable, OTRS_CONFIG_HttpType which is found in all forms of messaging used by the application, to build links to the tickets within your system.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'HttpType'} = 'http';
```

ScriptAlias

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'ScriptAlias'} = 'otrs/';
```

AdminEmail

Defines the system administrator's email address. It will be displayed in the error screens of the application.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

Organization

Company name which will be included in outgoing emails as an X-Header.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Organization'} = 'Example Company';
```

DefaultLanguage

Defines the default front-end language. All the possible values are determined by the available language files on the system (see the next setting).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'DefaultLanguage'} = 'en';
```

DefaultUsedLanguages

Defines all the languages that are available to the application. Specify only English names of languages here.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'DefaultUsedLanguages'} = {  
  'ar_SA' => 'Arabic (Saudi Arabia)',  
  'bg' => 'Bulgarian',  
  'ca' => 'Catalan',  
  'cs' => 'Czech',  
  'da' => 'Danish',  
  'de' => 'German',  
  'el' => 'Greek',  
  'en' => 'English (United States)',  
  'en_CA' => 'English (Canada)',  
  'en_GB' => 'English (United Kingdom)',  
  'es' => 'Spanish',  
  'es_CO' => 'Spanish (Colombia)',  
  'es_MX' => 'Spanish (Mexico)',  
  'et' => 'Estonian',  
  'fa' => 'Persian',  
  'fi' => 'Finnish',  
  'fr' => 'French',  
  'fr_CA' => 'French (Canada)',  
  'gl' => 'Galician',  
  'he' => 'Hebrew',  
  'hi' => 'Hindi',  
  'hr' => 'Croatian',  
  'hu' => 'Hungarian',  
  'id' => 'Indonesian',
```

```
'it' => 'Italian',
'ja' => 'Japanese',
'lt' => 'Lithuanian',
'lv' => 'Latvian',
'ms' => 'Malay',
'nb_NO' => 'Norwegian',
'nl' => 'Nederlands',
'pl' => 'Polish',
'pt' => 'Portuguese',
'pt_BR' => 'Portuguese (Brasil)',
'ru' => 'Russian',
'sk_SK' => 'Slovak',
'sl' => 'Slovenian',
'sr_Cyrl' => 'Serbian Cyrillic',
'sr_Latn' => 'Serbian Latin',
'sv' => 'Swedish',
'sw' => 'Swahili',
'th_TH' => 'Thai',
'tr' => 'Turkish',
'uk' => 'Ukrainian',
'vi_VN' => 'Vietnam',
'zh_CN' => 'Chinese (Simplified)',
'zh_TW' => 'Chinese (Traditional)'
};
```

DefaultUsedLanguagesNative

Defines all the languages that are available to the application. Specify only native names of languages here.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'DefaultUsedLanguagesNative'} = {
  'ar_SA' => 'العَرَبِيَّةُ',
  'bg' => 'Български',
  'ca' => 'Català',
  'cs' => 'Česky',
  'da' => 'Dansk',
  'de' => 'Deutsch',
  'el' => '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' => 'فارسی',
  'fi' => 'Suomi',
  'fr' => 'Français',
  'fr_CA' => 'Français (Canada)',
  'gl' => 'Galego',
  'he' => '#####',
  'hi' => '#####',
  'hr' => 'Hrvatski',
  'hu' => 'Magyar',
  'id' => 'Bahasa Indonesia',
  'it' => 'Italiano',
  'ja' => '日本語',
  'lt' => 'Lietuvių kalba',
  'lv' => 'Latvijas',
  'ms' => 'Melayu',
  'nb_NO' => 'Norsk bokmål',
  'nl' => 'Nederlandse',
  'pl' => 'Polski',
  'pt' => 'Português',
  'pt_BR' => 'Português Brasileiro',
  'ru' => 'Русский',
  'sk_SK' => 'Slovenčina',
  'sl' => 'Slovenščina',
```

```
'sr_Cyrl' => 'Српски',
'sr_Latn' => 'Srpski',
'sv' => 'Svenska',
'sw' => 'Kiswahili',
'th_TH' => '#####',
'tr' => 'Türkçe',
'uk' => 'Українська',
'vi_VN' => 'Việt Nam',
'zh_CN' => '简体中文',
'zh_TW' => '正體中文'
};
```

DefaultTheme

Defines the default front-end (HTML) theme to be used by the agents and customers. If you like, you can add your own theme. Please refer the administrator manual located at <http://otrs.github.io/doc/>.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'DefaultTheme'} = 'Standard';
```

DefaultTheme::HostBased

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

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

CheckMXRecord

Makes the application check the MX record of email addresses before sending an email or submitting a telephone or email ticket.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CheckMXRecord'} = '1';
```

CheckMXRecord::Nameserver

Defines the address of a dedicated DNS server, if necessary, for the "CheckMXRecord" look-ups.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

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

CheckEmailAddresses

Makes the application check the syntax of email addresses.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CheckEmailAddresses'} = '1';
```

CheckEmailValidAddress

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").

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CheckEmailValidAddress'} = '^(root@localhost|admin@localhost)$';
```

CheckEmailInvalidAddress

Defines a regular expression that filters all email addresses that should not be used in the application.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CheckEmailInvalidAddress'} = '@(example)\.\.(.|\.)$';
```

CGILogPrefix

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CGILogPrefix'} = 'OTRS-CGI';
```

DemoSystem

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'DemoSystem'} = '0';
```

OutOfOfficeMessageTemplate

Defines out of office message template. Two string parameters (%s) available: end date and number of days left.

Standardwert:

```
$Self->{'OutOfOfficeMessageTemplate'} = '*** out of office until %s (%s d left) ***';
```

SwitchToUser

Allows the administrators to login as other users, via the users administration panel.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SwitchToUser'} = '0';
```

SwitchToCustomer

Allows the administrators to login as other customers, via the customer user administration panel.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SwitchToCustomer'} = '0';
```

SwitchToCustomer::PermissionGroup

Specifies the group where the user needs rw permissions so that he can access the "SwitchToCustomer" feature.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SwitchToCustomer::PermissionGroup'} = 'admin';
```

NotificationSenderName

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 Notifications" otrs@your.example.com).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'NotificationSenderName'} = 'OTRS Notifications';
```

NotificationSenderEmail

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 Notifications" otrs@your.example.com). You can use the OTRS_CONFIG_FQDN variable as set in your configuration, or choose another email address.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'NotificationSenderEmail'} = 'otrs@<OTRS_CONFIG_FQDN>';
```

System::Customer::Permission

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'System::Customer::Permission'} = [  
  'ro',  
  'rw'  
];
```

LanguageDebug

Debugs the translation set. If this is set to "Yes" all strings (text) without translations are written to STDERR. This can be helpful when you are creating a new translation file. Otherwise, this option should remain set to "No".

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'LanguageDebug'} = '0';
```

Secure::DisableBanner

If enabled, the OTRS version tag will be removed from the Webinterface, the HTTP headers and the X-Headers of outgoing mails.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Secure::DisableBanner'} = '0';
```

Framework → Core::Cache

Cache::Module

Selects the cache backend to use.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Cache::Module'} = 'Kernel::System::Cache::FileStorable';
```

Cache::InMemory

Should the cache data be held in memory?

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Cache::InMemory'} = '1';
```

Cache::InBackend

Should the cache data be stored in the selected cache backend?

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Cache::InBackend'} = '1';
```

Cache::SubdirLevels

Specify how many sub directory levels to use when creating cache files. This should prevent too many cache files being in one directory.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Cache::SubdirLevels'} = '2';
```

Framework → Core::CustomerCompany

CustomerCompany::EventModulePost###2000-UpdateCustomerUsers

Event module that updates customer users after an update of the Customer.

Standardwert:

```
$Self->{'CustomerCompany::EventModulePost'}->{'2000-UpdateCustomerUsers'} = {
  'Event' => 'CustomerCompanyUpdate',
  'Module' => 'Kernel::System::CustomerCompany::Event::CustomerUserUpdate',
  'Transaction' => '0'
};
```

CustomerCompany::EventModulePost###2100-UpdateDynamicFieldName

Event module that updates customer company object name for dynamic fields.

Standardwert:

```
$Self->{'CustomerCompany::EventModulePost'}->{'2100-UpdateDynamicFieldObjectName'} = {
  'Event' => 'CustomerCompanyUpdate',
  'Module' => 'Kernel::System::CustomerCompany::Event::DynamicFieldObjectNameUpdate',
  'Transaction' => '0'
};
```

Framework → Core::CustomerUser

CustomerUser::EventModulePost###2100-UpdateSearchProfiles

Event module that updates customer user search profiles if login changes.

Standardwert:

```
$Self->{'CustomerUser::EventModulePost'}->{'2100-UpdateSearchProfiles'} = {
  'Event' => 'CustomerUserUpdate',
  'Module' => 'Kernel::System::CustomerUser::Event::SearchProfileUpdate',
  'Transaction' => '0'
};
```

CustomerUser::EventModulePost###2200-UpdateServiceMembership

Event module that updates customer user service membership if login changes.

Standardwert:

```
$Self->{'CustomerUser::EventModulePost'}->{'2200-UpdateServiceMembership'} = {
  'Event' => 'CustomerUserUpdate',
  'Module' => 'Kernel::System::CustomerUser::Event::ServiceMemberUpdate',
  'Transaction' => '0'
};
```

CustomerUser::EventModulePost###2000-UpdateDynamicFieldObjectName

Event module that updates customer user object name for dynamic fields.

Standardwert:

```
$Self->{'CustomerUser::EventModulePost'}->{'2000-UpdateDynamicFieldObjectName'} = {
  'Event' => 'CustomerUserUpdate',
  'Module' => 'Kernel::System::CustomerUser::Event::DynamicFieldObjectNameUpdate',
  'Transaction' => '0'
};
```

Framework → Core::Fetchmail

Fetchmail::Bin

Defines the fall-back path to open fetchmail binary. Note: The name of the binary needs to be 'fetchmail', if it is different please use a symbolic link.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Fetchmail::Bin'} = '/usr/bin/fetchmail';
```

Framework → Core::LinkObject

LinkObject::ViewMode

Determines the way the linked objects are displayed in each zoom mask.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'LinkObject::ViewMode'} = 'Simple';
```

LinkObject::Type###Normal

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'LinkObject::Type'}->{'Normal'} = {  
  'SourceName' => 'Normal',  
  'TargetName' => 'Normal'  
};
```

LinkObject::Type###ParentChild

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'LinkObject::Type'}->{'ParentChild'} = {  
  'SourceName' => 'Parent',  
  'TargetName' => 'Child'  
};
```

LinkObject::TypeGroup###0001

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.

Standardwert:

```
$Self->{'LinkObject::TypeGroup'}->{'0001'} = [  
  'Normal',  
  'ParentChild'  
];
```

Framework → Core::Log

LogModule

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'LogModule'} = 'Kernel::System::Log::SysLog';
```

LogModule::SysLog::Facility

If "SysLog" was selected for LogModule, a special log facility can be specified.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'LogModule::SysLog::Facility'} = 'user';
```

LogModule::SysLog::Charset

If "SysLog" was selected for LogModule, the charset that should be used for logging can be specified.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'LogModule::SysLog::Charset'} = 'utf-8';
```

LogModule::LogFile

If "file" was selected for LogModule, a logfile must be specified. If the file doesn't exist, it will be created by the system.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'LogModule::LogFile'} = '/tmp/otrs.log';
```

LogModule::LogFile::Date

Adds a suffix with the actual year and month to the OTRS log file. A logfile for every month will be created.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'LogModule::LogFile::Date'} = '0';
```

MinimumLogLevel

Set minimum loglevel. If you select 'error', just errors are logged. With 'debug' you get all logging messages.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'MinimumLogLevel'} = 'error';
```

Framework → Core::MIME-Viewer

MIME-Viewer###application/excel

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

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'MIME-Viewer'}->{'application/excel'} = 'xlhtml';
```

MIME-Viewer###application/msword

Specifies the path to the converter that allows the view of Microsoft Word files, in the web interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'MIME-Viewer'}->{'application/msword'} = 'wvWare';
```

MIME-Viewer###application/pdf

Specifies the path to the converter that allows the view of PDF documents, in the web interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'MIME-Viewer'}->{'application/pdf'} = 'pdftohtml -stdout -i';
```

MIME-Viewer###text/xml

Specifies the path to the converter that allows the view of XML files, in the web interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

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

Framework → Core::MirrorDB

Core::MirrorDB::DSN

OTRS can use one or more readonly mirror databases for expensive operations like fulltext search or statistics generation. Here you can specify the DSN for the first mirror database.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Core::MirrorDB::DSN'} = 'DBI:mysql:database=mirrordb;host=mirrordbhost';
```

Core::MirrorDB::User

Specify the username to authenticate for the first mirror database.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Core::MirrorDB::User'} = 'some_user';
```

Core::MirrorDB::Password

Specify the password to authenticate for the first mirror database.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

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

Core::MirrorDB::AdditionalMirrors###1

Configure any additional readonly mirror databases that you want to use.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Core::MirrorDB::AdditionalMirrors'}->{'1'} = {
  'DSN' => 'DBI:mysql:database=mirrordb;host=mirrordbhost',
  'Password' => 'some_password',
  'User' => 'some_user'
};
```

Core::MirrorDB::AdditionalMirrors###2

Configure any additional readonly mirror databases that you want to use.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Core::MirrorDB::AdditionalMirrors'}->{'2'} = {
  'DSN' => 'DBI:mysql:database=mirrordb;host=mirrordbhost',
  'Password' => 'some_password',
  'User' => 'some_user'
};
```

Core::MirrorDB::AdditionalMirrors###3

Configure any additional readonly mirror databases that you want to use.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Core::MirrorDB::AdditionalMirrors'}->{'3'} = {  
  'DSN' => 'DBI:mysql:database=mirrordb;host=mirrordbhost',  
  'Password' => 'some_password',  
  'User' => 'some_user'  
};
```

Core::MirrorDB::AdditionalMirrors###4

Configure any additional readonly mirror databases that you want to use.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Core::MirrorDB::AdditionalMirrors'}->{'4'} = {  
  'DSN' => 'DBI:mysql:database=mirrordb;host=mirrordbhost',  
  'Password' => 'some_password',  
  'User' => 'some_user'  
};
```

Core::MirrorDB::AdditionalMirrors###5

Configure any additional readonly mirror databases that you want to use.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Core::MirrorDB::AdditionalMirrors'}->{'5'} = {  
  'DSN' => 'DBI:mysql:database=mirrordb;host=mirrordbhost',  
  'Password' => 'some_password',  
  'User' => 'some_user'  
};
```

Framework → Core::OTRSBusiness

OTRSBusiness::ReleaseChannel

Specify the channel to be used to fetch OTRS Business Solution™ updates. Warning: Development releases might not be complete, your system might experience unrecoverable errors and on extreme cases could become unresponsive!

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'OTRSBusiness::ReleaseChannel'} = '1';
```

Framework → Core::PDF

PDF::LogoFile

Specifies the path of the file for the logo in the page header (gif|jpg|png, 700 x 100 pixel).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PDF::LogoFile'} = '<OTRS_CONFIG_Home>/var/logo-otrs.png';
```

PDF::PageSize

Defines the standard size of PDF pages.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PDF::PageSize'} = 'a4';
```

PDF::MaxPages

Defines the maximum number of pages per PDF file.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PDF::MaxPages'} = '100';
```

PDF::TTFontFile###Proportional

Defines the path and TTF-File to handle proportional font in PDF documents.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PDF::TTFontFile'}->{'Proportional'} = 'DejaVuSans.ttf';
```

PDF::TTFontFile###ProportionalBold

Defines the path and TTF-File to handle bold proportional font in PDF documents.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PDF::TTFontFile'}->{'ProportionalBold'} = 'DejaVuSans-Bold.ttf';
```

PDF::TTFontFile###ProportionalItalic

Defines the path and TTF-File to handle italic proportional font in PDF documents.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

PDF::TTFontFile###ProportionalBoldItalic

Defines the path and TTF-File to handle bold italic proportional font in PDF documents.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PDF::TTFontFile'}->{'ProportionalBoldItalic'} = 'DejaVuSans-BoldOblique.ttf';
```

PDF::TTFontFile###Monospaced

Defines the path and TTF-File to handle monospaced font in PDF documents.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PDF::TTFontFile'}->{'Monospaced'} = 'DejaVuSansMono.ttf';
```

PDF::TTFontFile###MonospacedBold

Defines the path and TTF-File to handle bold monospaced font in PDF documents.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PDF::TTFontFile'}->{'MonospacedBold'} = 'DejaVuSansMono-Bold.ttf';
```

PDF::TTFontFile###MonospacedItalic

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PDF::TTFontFile'}->{'MonospacedItalic'} = 'DejaVuSansMono-Oblique.ttf';
```

PDF::TTFontFile###MonospacedBoldItalic

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PDF::TTFontFile'}->{'MonospacedBoldItalic'} = 'DejaVuSansMono-BoldOblique.ttf';
```

Framework → Core::Package

Package::FileUpload

Enables file upload in the package manager frontend.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Package::FileUpload'} = '1';
```

Package::RepositoryRoot

Defines the location to get online repository list for additional packages. The first available result will be used.

Standardwert:

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

Package::RepositoryList

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".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

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

Package::RepositoryAccessRegExp

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Package::RepositoryAccessRegExp'} = '127\\.0\\.0\\.1';
```

Package::**Timeout**

Sets the timeout (in seconds) for package downloads. Overwrites "WebUserAgent::Timeout".

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Package::Timeout'} = '120';
```

Package::**Proxy**

Fetches packages via proxy. Overwrites "WebUserAgent::Proxy".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Package::Proxy'} = 'http://proxy.sn.no:8001/';
```

Package::**AllowLocalModifications**

If this setting is active, local modifications will not be highlighted as errors in the package manager and support data collector.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Package::AllowLocalModifications'} = '0';
```

Package::**ShowFeatureAddons**

Toggles display of OTRS FeatureAddons list in PackageManager.

Standardwert:

```
$Self->{'Package::ShowFeatureAddons'} = '1';
```

Package::**EventModulePost###9000-SupportDataSend**

Package event module file a scheduler task for update registration.

Standardwert:

```
$Self->{'Package::EventModulePost'}->{'9000-SupportDataSend'} = {  
  'Event' => '(PackageInstall|PackageReinstall|PackageUpgrade|PackageUninstall)',  
  'Module' => 'Kernel::System::Package::Event::SupportDataSend',  
  'Transaction' => '1'  
};
```

Framework → Core::**PerformanceLog**

PerformanceLog

Enables performance log (to log the page response time). It will affect the system performance. Frontend::Module###AdminPerformanceLog must be enabled.

Standardwert:

```
$Self->{'PerformanceLog'} = '0';
```

PerformanceLog::**File**

Specifies the path of the file for the performance log.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PerformanceLog::File'} = '<OTRS_CONFIG_Home>/var/log/Performance.log';
```

PerformanceLog::FileMax

Defines the maximum size (in MB) of the log file.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PerformanceLog::FileMax'} = '25';
```

Framework → Core::ReferenceData

ReferenceData::OwnCountryList

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'ReferenceData::OwnCountryList'} = {  
  'AT' => 'Austria',  
  'CH' => 'Switzerland',  
  'DE' => 'Germany'  
};
```

Framework → Core::SOAP

SOAP::User

Defines the username to access the SOAP handle (bin/cgi-bin/rpc.pl).

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'SOAP::User'} = 'some_user';
```

SOAP::Password

Defines the password to access the SOAP handle (bin/cgi-bin/rpc.pl).

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'SOAP::Password'} = 'some_pass';
```

SOAP::Keep-Alive

Enable keep-alive connection header for SOAP responses.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SOAP::Keep-Alive'} = '0';
```

Framework → Core::Sendmail

SendmailModule

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) mail-server. "DoNotSendEmail" doesn't send emails and it is useful for test systems.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

SendmailModule::CMD

If "Sendmail" was selected as SendmailModule, the location of the sendmail binary and the needed options must be specified.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SendmailModule::CMD'} = '/usr/sbin/sendmail -i -f';
```

SendmailModule::Host

If any of the "SMTP" mechanisms was selected as SendmailModule, the mailhost that sends out the mails must be specified.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SendmailModule::Host'} = 'mail.example.com';
```

SendmailModule::Port

If any of the "SMTP" mechanisms was selected as SendmailModule, the port where your mailserver is listening for incoming connections must be specified.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'SendmailModule::Port'} = '25';
```

SendmailModule::AuthUser

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

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'SendmailModule::AuthUser'} = 'MailserverLogin';
```

SendmailModule::AuthPassword

If any of the "SMTP" mechanisms was selected as SendmailModule, and authentication to the mail server is needed, a password must be specified.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'SendmailModule::AuthPassword'} = 'MailserverPassword';
```

SendmailBcc

Sends all outgoing email via bcc to the specified address. Please use this only for backup reasons.

Standardwert:

```
$Self->{'SendmailBcc'} = '';
```

SendmailEnvelopeFrom

If set, this address is used as envelope sender in outgoing messages (not notifications - see below). If no address is specified, the envelope sender is equal to queue e-mail address.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'SendmailEnvelopeFrom'} = '';
```

SendmailNotificationEnvelopeFrom

If set, this address is used as envelope sender header in outgoing notifications. If no address is specified, the envelope sender header is empty.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

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

SendmailEncodingForce

Forces encoding of outgoing emails (7bit|8bit|quoted-printable|base64).

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'SendmailEncodingForce'} = 'base64';
```

Sendmail::DefaultHeaders

Defines default headers for outgoing emails.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Sendmail::DefaultHeaders'} = {  
  'Auto-Submitted:' => 'auto-generated',  
  'Precedence:' => 'bulk'  
};
```

Framework → Core::Session

SessionModule

Defines the module used to store the session data. With "DB" the frontend server can be splitted from the db server. "FS" is faster.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SessionModule'} = 'Kernel::System::AuthSession::DB';
```

SessionName

Defines the name of the session key. E.g. Session, SessionID or OTRS.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SessionName'} = 'OTRSAgentInterface';
```

CustomerPanelSessionName

Defines the name of the key for customer sessions.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerPanelSessionName'} = 'OTRSCustomerInterface';
```

SessionCheckRemoteIP

Turns on the remote ip address check. It should be set to "No" if the application is used, for example, via a proxy farm or a dialup connection, because the remote ip address is mostly different for the requests.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SessionCheckRemoteIP'} = '1';
```

SessionDeleteIfNotRemoteID

Deletes a session if the session id is used with an invalid remote IP address.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SessionDeleteIfNotRemoteID'} = '1';
```

SessionMaxTime

Defines the maximal valid time (in seconds) for a session id.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SessionMaxTime'} = '57600';
```

SessionMaxIdleTime

Sets the inactivity time (in seconds) to pass before a session is killed and a user is logged out.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

SessionActiveTime

Sets the time (in seconds) a user is marked as active (minimum active time is 300 seconds).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SessionActiveTime'} = '600';
```

SessionDeleteIfTimeToOld

Deletes requested sessions if they have timed out.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SessionDeleteIfTimeToOld'} = '1';
```

SessionUseCookie

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SessionUseCookie'} = '1';
```

SessionUseCookieAfterBrowserClose

Stores cookies after the browser has been closed.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SessionUseCookieAfterBrowserClose'} = '0';
```

SessionCSRFProtection

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SessionCSRFProtection'} = '1';
```

AgentSessionLimitPriorWarning

Sets the maximum number of active agents within the timespan defined in `SessionActiveTime` before a prior warning will be visible for the logged in agents.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'AgentSessionLimitPriorWarning'} = '90';
```

AgentSessionLimit

Sets the maximum number of active agents within the timespan defined in `SessionActiveTime`.

Standardwert:

```
$Self->{'AgentSessionLimit'} = '100';
```

AgentSessionPerUserLimit

Sets the maximum number of active sessions per agent within the timespan defined in `SessionActiveTime`.

Standardwert:

```
$Self->{'AgentSessionPerUserLimit'} = '20';
```

CustomerSessionLimit

Sets the maximum number of active customers within the timespan defined in `SessionActiveTime`.

Standardwert:

```
$Self->{'CustomerSessionLimit'} = '100';
```

CustomerSessionPerUserLimit

Sets the maximum number of active sessions per customers within the timespan defined in `SessionActiveTime`.

Standardwert:

```
$Self->{'CustomerSessionPerUserLimit'} = '20';
```

SessionDir

If "FS" was selected for SessionModule, a directory where the session data will be stored must be specified.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SessionDir'} = '<OTRS_CONFIG_Home>/var/sessions';
```

SessionTable

If "DB" was selected for SessionModule, a table in database where session data will be stored must be specified.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SessionTable'} = 'sessions';
```

Framework → Core::SpellChecker

SpellChecker

Enables spell checker support.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SpellChecker'} = '0';
```

SpellCheckerBin

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SpellCheckerBin'} = '/usr/bin/ispell';
```

SpellCheckerDictDefault

Defines the default spell checker dictionary.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SpellCheckerDictDefault'} = 'english';
```

SpellCheckerIgnore

Defines a default list of words, that are ignored by the spell checker.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SpellCheckerIgnore'} = [  
  'www',  
  'webmail',  
  'https',  
  'http',  
  'html',  
  'rfc'  
];
```

Framework → Core::Stats

Stats::StatsHook

Sets the stats hook.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Stats::StatsHook'} = 'Stat#';
```

Stats::StatsStartNumber

Start number for statistics counting. Every new stat increments this number.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Stats::StatsStartNumber'} = '10000';
```

Stats::MaxXaxisAttributes

Defines the default maximum number of X-axis attributes for the time scale.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Stats::MaxXaxisAttributes'} = '1000';
```

Framework → Core::Time

TimeInputFormat

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeInputFormat'} = 'Option';
```

TimeShowAlwaysLong

Shows time in long format (days, hours, minutes), if set to "Yes"; or in short format (days, hours), if set to "No".

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeShowAlwaysLong'} = '0';
```

TimeZone

This setting is deprecated. Set OTRSTimeZone instead.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'TimeZone'} = '+0';
```

OTRSTimeZone

Sets the time zone being used internally by OTRS to e. g. store dates and times in the database. **WARNING:** This setting must not be changed once set and tickets or any other data containing date/time have been created.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'OTRSTimeZone'} = 'UTC';
```

UserDefaultTimeZone

Sets the time zone that will be assigned to newly created users and will be used for users that haven't yet set a time zone. This is the time zone being used as default to convert date and time between the OTRS time zone and the user's time zone.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'UserDefaultTimeZone'} = 'UTC';
```

ShowUserTimeZoneSelectionNotification

If enabled, users that haven't selected a time zone yet will be notified to do so. Note: Notification will not be shown if (1) user has not yet selected a time zone and (2) OTRSTimeZone and UserDefaultTimeZone do match and (3) are not set to UTC.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'ShowUserTimeZoneSelectionNotification'} = '1';
```

MaximumCalendarNumber

Maximum Number of a calendar shown in a dropdown.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'MaximumCalendarNumber'} = '50';
```

CalendarWeekDayStart

Define the start day of the week for the date picker.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CalendarWeekDayStart'} = '1';
```

TimeVacationDays

Adds the permanent vacation days. Please use single digit pattern for numbers from 1 to 9 (instead of 01 - 09).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$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'  
}  
};
```

TimeVacationDaysOneTime

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

TimeWorkingHours

Defines the hours and week days to count the working time.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

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

TimeShowCompleteDescription

Shows time use complete description (days, hours, minutes), if set to "Yes"; or just first letter (d, h, m), if set to "No".

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

Framework → Core::Time::Calendar1

TimeZone::Calendar1Name

Defines the name of the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeZone::Calendar1Name'} = 'Calendar Name 1';
```

TimeZone::Calendar1

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

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'TimeZone::Calendar1'} = 'UTC';
```


CalendarWeekDayStart::Calendar1

Define the start day of the week for the date picker for the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CalendarWeekDayStart::Calendar1'} = '1';
```

TimeVacationDays::Calendar1

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

TimeVacationDaysOneTime::Calendar1

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).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

TimeWorkingHours::Calendar1

Defines the hours and week days of the indicated calendar, to count the working time.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

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

Framework → Core::Time::Calendar2

TimeZone::Calendar2Name

Defines the name of the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeZone::Calendar2Name'} = 'Calendar Name 2';
```

TimeZone::Calendar2

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

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'TimeZone::Calendar2'} = 'UTC';
```

CalendarWeekDayStart::Calendar2

Define the start day of the week for the date picker for the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CalendarWeekDayStart::Calendar2'} = '1';
```

TimeVacationDays::Calendar2

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$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'  
  }  
};
```

TimeVacationDaysOneTime::Calendar2

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).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeVacationDaysOneTime::Calendar2'} = {  
  '2004' => {  
    '1' => {  
      '1' => 'test'  
    }  
  }  
};
```

TimeWorkingHours::Calendar2

Defines the hours and week days of the indicated calendar, to count the working time.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeWorkingHours::Calendar2'} = {  
  'Fri' => [  
    '8',  
    '9',  
    '10',  
    '11',  
    '12',  
    '13',  
    '14',  
    '15',  
    '16',  
    '17',  
    '18',  
    '19',  
    '20'  
  ],  
  'Mon' => [  
    '8',  
    '9',  
    '10',  
    '11',  
    '12',  
    '13',  
    '14',  
    '15',  
    '16',  
    '17',  
    '18',  
    '19',  
    '20'  
  ],  
  'Sat' => [],  
  'Sun' => [],  
  'Thu' => [  
    '8',  
    '9',  
    '10',  
    '11',  
    '12',  
    '13',  
    '14',  
    '15',  
    '16',  
    '17',  
    '18',  
    '19',  
    '20'  
  ],  
  'Tue' => [  
    '8',  
    '9',  
    '10',  
    '11',  
    '12',  
    '13',  
    '14',  
    '15',  
    '16',  
    '17',  
    '18',  
    '19',  
    '20'  
  ],  
  'Wed' => [  
    '8',  
    '9',  
    '10',  
    '11',  
  ]  
}
```

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

Framework → Core::Time::Calendar3

TimeZone::Calendar3Name

Defines the name of the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeZone::Calendar3Name'} = 'Calendar Name 3';
```

TimeZone::Calendar3

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

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'TimeZone::Calendar3'} = 'UTC';
```

CalendarWeekDayStart::Calendar3

Define the start day of the week for the date picker for the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CalendarWeekDayStart::Calendar3'} = '1';
```

TimeVacationDays::Calendar3

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$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'
  }
};
```

TimeVacationDaysOneTime::Calendar3

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).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeVacationDaysOneTime::Calendar3'} = {  
  '2004' => {  
    '1' => {  
      '1' => 'test'  
    }  
  }  
};
```

TimeWorkingHours::Calendar3

Defines the hours and week days of the indicated calendar, to count the working time.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

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

Framework → Core::Time::Calendar4

TimeZone::Calendar4Name

Defines the name of the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeZone::Calendar4Name'} = 'Calendar Name 4';
```

TimeZone::Calendar4

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

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'TimeZone::Calendar4'} = 'UTC';
```

CalendarWeekDayStart::Calendar4

Define the start day of the week for the date picker for the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CalendarWeekDayStart::Calendar4'} = '1';
```

TimeVacationDays::Calendar4

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$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'
}
};
```

TimeVacationDaysOneTime::Calendar4

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).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

TimeWorkingHours::Calendar4

Defines the hours and week days of the indicated calendar, to count the working time.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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



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

Framework → Core::Time::Calendar5

TimeZone::Calendar5Name

Defines the name of the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeZone::Calendar5Name'} = 'Calendar Name 5';
```

TimeZone::Calendar5

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

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'TimeZone::Calendar5'} = 'UTC';
```

CalendarWeekDayStart::Calendar5

Define the start day of the week for the date picker for the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CalendarWeekDayStart::Calendar5'} = '1';
```

TimeVacationDays::Calendar5

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$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'  
  }  
};
```

TimeVacationDaysOneTime::Calendar5

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).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

TimeWorkingHours::Calendar5

Defines the hours and week days of the indicated calendar, to count the working time.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeWorkingHours::Calendar5'} = {  
  'Fri' => [  
    '8',  
    '9',  
    '10',  
    '11',  
    '12',  
    '13',  
    '14',  
    '15',  
    '16',  
    '17',  
    '18',  
    '19',  
    '20'  
  ]  
};
```

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

Framework → Core::Time::Calendar6

TimeZone::Calendar6Name

Defines the name of the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeZone::Calendar6Name'} = 'Calendar Name 6';
```

TimeZone::Calendar6

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

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'TimeZone::Calendar6'} = 'UTC';
```

CalendarWeekDayStart::Calendar6

Define the start day of the week for the date picker for the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CalendarWeekDayStart::Calendar6'} = '1';
```

TimeVacationDays::Calendar6

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeVacationDays::Calendar6'} = {
  '1' => {
    '1' => 'New Year\'s Day'
  },
  '12' => {
    '24' => 'Christmas Eve',
    '25' => 'First Christmas Day',
    '26' => 'Second Christmas Day',
    '31' => 'New Year\'s Eve'
  },
  '5' => {
    '1' => 'International Workers\' Day'
  }
};
```

TimeVacationDaysOneTime::Calendar6

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).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

TimeWorkingHours::Calendar6

Defines the hours and week days of the indicated calendar, to count the working time.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeWorkingHours::Calendar6'} = {  
  'Fri' => [  
    '8',  
    '9',  
    '10',  
    '11',  
    '12',  
    '13',  
    '14',  
    '15',  
    '16',  
    '17',  
    '18',  
    '19',  
    '20'  
  ],  
  'Mon' => [  
    '8',  
    '9',  
    '10',  
    '11',  
    '12',  
    '13',  
    '14',  
    '15',  
    '16',  
    '17',  
    '18',  
    '19',  
    '20'  
  ],  
  'Sat' => [],  
  'Sun' => [],  
  'Thu' => [  
    '8',  
    '9',  
    '10',  
    '11',  
    '12',  
    '13',  
    '14',  
    '15',  
    '16',  
    '17',  
    '18',  
    '19',  
    '20'  
  ],  
  'Tue' => [  
    '8',  
    '9',  
    '10',  
    '11',  
    '12',  
    '13',  
    '14',  
    '15',  
    '16',  
    '17',  
    '18',  
    '19',  
    '20'  
  ],  
  'Wed' => [  
    '8',  
    '9',  
    '10',  
    '11',
```

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

Framework → Core::Time::Calendar7

TimeZone::Calendar7Name

Defines the name of the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeZone::Calendar7Name'} = 'Calendar Name 7';
```

TimeZone::Calendar7

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

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'TimeZone::Calendar7'} = 'UTC';
```

CalendarWeekDayStart::Calendar7

Define the start day of the week for the date picker for the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CalendarWeekDayStart::Calendar7'} = '1';
```

TimeVacationDays::Calendar7

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$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'
  }
};
```

TimeVacationDaysOneTime::Calendar7

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).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeVacationDaysOneTime::Calendar7'} = {  
  '2004' => {  
    '1' => {  
      '1' => 'test'  
    }  
  }  
};
```

TimeWorkingHours::Calendar7

Defines the hours and week days of the indicated calendar, to count the working time.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeWorkingHours::Calendar7'} = {  
  'Fri' => [  
    '8',  
    '9',  
    '10',  
    '11',  
    '12',  
    '13',  
    '14',  
    '15',  
    '16',  
    '17',  
    '18',  
    '19',  
    '20'  
  ],  
  'Mon' => [  
    '8',  
    '9',  
    '10',  
    '11',  
    '12',  
    '13',  
    '14',  
    '15',  
    '16',  
    '17',  
    '18',  
    '19',  
    '20'  
  ],  
  'Sat' => [],  
  'Sun' => [],  
  'Thu' => [  
    '8',  
    '9',  
    '10',  
    '11',  
    '12',  
    '13',  
    '14',  
    '15',  
    '16',  
    '17',  
    '18',  
    '19',  
    '20'  
  ],  
  'Tue' => [  
    '8',  
    '9',  
    '10',
```

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

Framework → Core::Time::Calendar8

TimeZone::Calendar8Name

Defines the name of the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

TimeZone::Calendar8

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

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'TimeZone::Calendar8'} = 'UTC';
```

CalendarWeekDayStart::Calendar8

Define the start day of the week for the date picker for the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CalendarWeekDayStart::Calendar8'} = '1';
```

TimeVacationDays::Calendar8

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$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'
}
};
```

TimeVacationDaysOneTime::Calendar8

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).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

TimeWorkingHours::Calendar8

Defines the hours and week days of the indicated calendar, to count the working time.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeWorkingHours::Calendar8'} = {
  'Fri' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Mon' => [
    '8',
    '9',
    '10',
    '11',
    '12',
    '13',
    '14',
    '15',
    '16',
    '17',
    '18',
    '19',
    '20'
  ],
  'Sat' => [],
  'Sun' => [],
  'Thu' => [
```

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

Framework → Core::Time::Calendar9

TimeZone::Calendar9Name

Defines the name of the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeZone::Calendar9Name'} = 'Calendar Name 9';
```

TimeZone::Calendar9

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

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'TimeZone::Calendar9'} = 'UTC';
```

CalendarWeekDayStart::Calendar9

Define the start day of the week for the date picker for the indicated calendar.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CalendarWeekDayStart::Calendar9'} = '1';
```

TimeVacationDays::Calendar9

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$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'
  }
};
```

TimeVacationDaysOneTime::Calendar9

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).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TimeVacationDaysOneTime::Calendar9'} = {
  '2004' => {
    '1' => {
      '1' => 'test'
    }
  }
};
```

TimeWorkingHours::Calendar9

Defines the hours and week days of the indicated calendar, to count the working time.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

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

Framework → Core::Web

Frontend::WebPath

Defines the URL base path of icons, CSS and Java Script.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::WebPath'} = '/otrs-web/';
```

Frontend::ImagePath

Defines the URL image path of icons for navigation.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::ImagePath'} = '<OTRS_CONFIG_Frontend::WebPath>skins/Agent/default/img/';
```

Frontend::CSSPath

Defines the URL CSS path.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::CSSPath'} = '<OTRS_CONFIG_Frontend::WebPath>css/';
```

Frontend::JavaScriptPath

Defines the URL java script path.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::JavaScriptPath'} = '<OTRS_CONFIG_Frontend::WebPath>js/';
```

Frontend::RichText

Uses richtext for viewing and editing: articles, salutations, signatures, standard templates, auto responses and notifications.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::RichText'} = '1';
```

Frontend::RichTextPath

Defines the URL rich text editor path.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::RichTextPath'} = '<OTRS_CONFIG_Frontend::WebPath>js/thirdparty/ckeditor-4.5.11/';
```

Frontend::RichTextWidth

Defines the width for the rich text editor component. Enter number (pixels) or percent value (relative).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::RichTextWidth'} = '620';
```

Frontend::RichTextHeight

Defines the height for the rich text editor component. Enter number (pixels) or percent value (relative).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::RichTextHeight'} = '320';
```

Frontend::RichText::DefaultCSS

Defines the default CSS used in rich text editors.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::RichText::DefaultCSS'} = 'font-family:Geneva,Helvetica,Arial,sans-serif; font-size: 12px;';
```

Frontend::RichText::EnhancedMode

Defines if the enhanced mode should be used (enables use of table, replace, subscript, superscript, paste from word, etc.).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::RichText::EnhancedMode'} = '0';
```

Frontend::RichText::EnhancedMode::Customer

Defines if the enhanced mode should be used (enables use of table, replace, subscript, superscript, paste from word, etc.) in customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::RichText::EnhancedMode::Customer'} = '0';
```

DisableIFrameOriginRestricted

Disable HTTP header "X-Frame-Options: SAMEORIGIN" to allow OTRS to be included as an IFrame in other websites. Disabling this HTTP header can be a security issue! Only disable it, if you know what you are doing!

Standardwert:

```
$Self->{'DisableIFrameOriginRestricted'} = '0';
```

DisableContentSecurityPolicy

Disable HTTP header "Content-Security-Policy" to allow loading of external script contents. Disabling this HTTP header can be a security issue! Only disable it, if you know what you are doing!

Standardwert:

```
$Self->{'DisableContentSecurityPolicy'} = '0';
```

DefaultViewNewLine

Automated line break in text messages after x number of chars.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

DefaultViewLines

Sets the number of lines that are displayed in text messages (e.g. ticket lines in the QueueZoom).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'DefaultViewLines'} = '6000';
```

Frontend::MenuDragDropEnabled

Turns on drag and drop for the main navigation.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::MenuDragDropEnabled'} = '1';
```

AttachmentDownloadType

Allows choosing between showing the attachments of a ticket in the browser (inline) or just make them downloadable (attachment).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'AttachmentDownloadType'} = 'attachment';
```

WebMaxFileUpload

Defines the maximal size (in bytes) for file uploads via the browser. Warning: Setting this option to a value which is too low could cause many masks in your OTRS instance to stop working (probably any mask which takes input from the user).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'WebMaxFileUpload'} = '24000000';
```

WebUploadCacheModule

Selects the module to handle uploads via the web interface. "DB" stores all uploads in the database, "FS" uses the file system.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'WebUploadCacheModule'} = 'Kernel::System::Web::UploadCache::DB';
```

Frontend::Output::FilterText###AAAURL

Defines the filter that processes the text in the articles, in order to highlight URLs.

Standardwert:

```
$Self->{'Frontend::Output::FilterText'}->{'AAAURL'} = {  
  'Module' => 'Kernel::Output::HTML::FilterText::URL',  
  'Templates' => {  
    'AgentTicketZoom' => '1'  
  }  
};
```

Frontend::Themes

Activates the available themes on the system. Value 1 means active, 0 means inactive.

Standardwert:

```
$Self->{'Frontend::Themes'} = {  
  'Lite' => '0',  
  'Standard' => '1'  
};
```

Frontend::Output::FilterText###OutputFilterTextAutoLink

Defines a filter to process the text in the articles, in order to highlight predefined keywords.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::Output::FilterText'}->{'OutputFilterTextAutoLink'} = {
  'Module' => 'Kernel::Output::HTML::FilterText::AutoLink',
  'Templates' => {
    'AgentTicketZoom' => '1'
  }
};
```

Frontend::Output::OutputFilterTextAutoLink###CVE

Defines a filter for html output to add links behind CVE numbers. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possibility is to insert the link to the image.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::Output::OutputFilterTextAutoLink'}->{'CVE'} = {
  'RegExp' => [
    '(CVE|CAN)\-\(\d{3,4})\-\(\d{2,})'
  ],
  'Templates' => {
    'AgentTicketZoom' => '1'
  },
  'URL1' => {
    'Description' => 'Mitre',
    'Image' => 'http://cve.mitre.org/favicon.ico',
    'Target' => '_blank',
    'URL' => 'http://cve.mitre.org/cgi-bin/cvename.cgi?name=<MATCH1>-<MATCH2>-<MATCH3>'
  },
  'URL2' => {
    'Description' => 'Google',
    'Image' => 'http://www.google.de/favicon.ico',
    'Target' => '_blank',
    'URL' => 'http://google.com/search?q=<MATCH1>-<MATCH2>-<MATCH3>'
  },
  'URL3' => {
    'Description' => 'US-CERT NVD',
    'Image' => 'http://nvd.nist.gov/favicon.ico',
    'Target' => '_blank',
    'URL' => 'http://nvd.nist.gov/nvd.cfm?cvename=<MATCH1>-<MATCH2>-<MATCH3>'
  }
};
```

Frontend::Output::OutputFilterTextAutoLink###Bugtraq

Defines a filter for html output to add links behind bugtraq numbers. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possibility is to insert the link to the image.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

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



```
'
  '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>'
}
}
};
```

Frontend::Output::OutputFilterTextAutoLink###MSBulletins

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

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$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>'
  }
}
};
```

Frontend::Output::OutputFilterTextAutoLink###Setting1

Define a filter for html output to add links behind a defined string. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possibility is to insert the link to the image.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$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'
}
};
```

Frontend::Output::OutputFilterTextAutoLink###Setting2

Defines a filter for html output to add links behind a defined string. The element Image allows two input kinds. At once the name of an image (e.g. faq.png). In this case the OTRS image path will be used. The second possibility is to insert the link to the image.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$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'
  }
};
```

Loader::Enabled::CSS

If enabled, OTRS will deliver all CSS files in minified form.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

Loader::Enabled::JS

If enabled, OTRS will deliver all JavaScript files in minified form.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

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

List of CSS files to always be loaded for the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$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.Login.css',
  'Core.Widget.css',
  'Core.WidgetMenu.css',
  'Core.TicketDetail.css',
  'Core.Tooltip.css',
  'Core.Dialog.css',
  'Core.InputFields.css',
  'Core.Print.css',
  'Core.Animations.css'
];
```

Loader::Agent::ResponsiveCSS###000-Framework

List of responsive CSS files to always be loaded for the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Loader::Agent::ResponsiveCSS'}->{'000-Framework'} = [
  'Core.Responsive.css'
];
```

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

List of JS files to always be loaded for the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Loader::Agent::CommonJS'}->{'000-Framework'} = [
  'thirdparty/jquery-3.1.1/jquery.js',
  'thirdparty/jquery-browser-detection/jquery-browser-detection.js',
  'thirdparty/jquery-ui-1.12.1/jquery-ui.js',
  'thirdparty/jquery-ui-touch-punch-0.2.3/jquery.ui.touch-punch.js',
  'thirdparty/jquery-validate-1.14.0/jquery.validate.js',
  'thirdparty/stacktrace-0.6.4/stacktrace.js',
  'thirdparty/jquery-pubsub/pubsub.js',
  'thirdparty/jquery-jstree-3.1.1/jquery.jstree.js',
  'thirdparty/nunjucks-2.4.2/nunjucks.js',
  'Core.Init.js',
  'Core.JavaScriptEnhancements.js',
  'Core.Debug.js',
  'Core.Exception.js',
  'Core.Data.js',
  'Core.Config.js',
  'Core.Language.js',
  'Core.Template.js',
  'Core.JSON.js',
  'Core.App.js',
  'Core.App.Responsive.js',
  'Core.AJAX.js',
  'Core.UI.js',
  'Core.UI.InputFields.js',
  'Core.UI.Accordion.js',
];
```

```
'Core.UI.Datepicker.js',
'Core.UI.DnD.js',
'Core.UI.Floater.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.UI.TreeSelection.js',
'Core.UI.Autocomplete.js',
'Core.Form.js',
'Core.Form.ErrorTooltips.js',
'Core.Form.Validate.js',
'Core.Agent.js',
'Core.Agent.Search.js',
'Core.Agent.CustomerInformationCenterSearch.js',
'Core.Agent.Header.js',
'Core.UI.Notification.js',
'Core.Agent.Responsive.js'
];
```

Loader::Agent::CommonJS###001-JQueryMigrate

List of JS files to always be loaded for the agent interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Loader::Agent::CommonJS'}->{'001-JQueryMigrate'} = [
'thirdparty/jquery-migrate-3.0.0/jquery-migrate.js'
];
```

Loader::Agent::CommonJS###100-CKEditor

List of JS files to always be loaded for the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Loader::Agent::CommonJS'}->{'100-CKEditor'} = [
'thirdparty/ckeditor-4.5.11/ckeditor.js'
];
```

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

List of CSS files to always be loaded for the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Loader::Customer::CommonCSS'}->{'000-Framework'} = [
'Core.Reset.css',
'Core.Default.css',
'Core.Form.css',
'Core.Dialog.css',
'Core.Tooltip.css',
'Core.Login.css',
'Core.Control.css',
'Core.Table.css',
'Core.TicketZoom.css',
'Core.InputFields.css',
'Core.Print.css',
'Core.Animations.css'
];
```

Loader::Customer::ResponsiveCSS###000-Framework

List of responsive CSS files to always be loaded for the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Loader::Customer::ResponsiveCSS'}->{'000-Framework'} = [  
  'Core.Responsive.css'  
];
```

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

List of JS files to always be loaded for the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Loader::Customer::CommonJS'}->{'000-Framework'} = [  
  'thirdparty/jquery-3.1.1/jquery.js',  
  'thirdparty/jquery-browser-detection/jquery-browser-detection.js',  
  'thirdparty/jquery-validate-1.14.0/jquery.validate.js',  
  'thirdparty/jquery-ui-1.12.1/jquery-ui.js',  
  'thirdparty/stacktrace-0.6.4/stacktrace.js',  
  'thirdparty/jquery-pubsub/pubsub.js',  
  'thirdparty/jquery-jstree-3.1.1/jquery.jstree.js',  
  'thirdparty/nunjucks-2.4.2/nunjucks.js',  
  'Core.Init.js',  
  'Core.Debug.js',  
  'Core.Exception.js',  
  'Core.Data.js',  
  'Core.JSON.js',  
  'Core.JavaScriptEnhancements.js',  
  'Core.Config.js',  
  'Core.Language.js',  
  'Core.Template.js',  
  'Core.App.js',  
  'Core.App.Responsive.js',  
  'Core.AJAX.js',  
  'Core.UI.js',  
  'Core.UI.InputFields.js',  
  'Core.UI.Accessibility.js',  
  'Core.UI.Dialog.js',  
  'Core.UI.RichTextEditor.js',  
  'Core.UI.Datepicker.js',  
  'Core.UI.Popup.js',  
  'Core.UI.TreeSelection.js',  
  'Core.UI.Autocomplete.js',  
  'Core.Form.js',  
  'Core.Form.ErrorTooltips.js',  
  'Core.Form.Validate.js',  
  'Core.Customer.js',  
  'Core.Customer.Responsive.js'  
];
```

Loader::Customer::CommonJS###001-JQueryMigrate

List of JS files to always be loaded for the customer interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Loader::Customer::CommonJS'}->{'001-JQueryMigrate'} = [  
  'thirdparty/jquery-migrate-3.0.0/jquery-migrate.js'  
];
```

Loader::Customer::CommonJS###100-CKEditor

List of JS files to always be loaded for the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Loader::Customer::CommonJS'}->{'100-CKEditor'} = [  
  'thirdparty/ckeditor-4.5.11/ckeditor.js'  
];
```

Framework → Core::WebUserAgent

WebUserAgent::Timeout

Sets the timeout (in seconds) for http/ftp downloads.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'WebUserAgent::Timeout'} = '15';
```

WebUserAgent::Proxy

Defines the connections for http/ftp, via a proxy.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'WebUserAgent::Proxy'} = 'http://proxy.sn.no:8001/';
```

WebUserAgent::DisableSSLVerification

Turns off SSL certificate validation, for example if you use a transparent HTTPS proxy. Use at your own risk!

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'WebUserAgent::DisableSSLVerification'} = '0';
```

Framework → Crypt::PGP

PGP

Enables PGP support. When PGP support is enabled for signing and encrypting mail, it is HIGHLY recommended that the web server runs as the OTRS user. Otherwise, there will be problems with the privileges when accessing .gnupg folder.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PGP'} = '0';
```

PGP::Bin

Defines the path to PGP binary.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PGP::Bin'} = '/usr/bin/gpg';
```

PGP::Options

Sets the options for PGP binary.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PGP::Options'} = '--homedir /opt/otrs/.gnupg/ --batch --no-tty --yes';
```

PGP::Options::DigestPreference

Sets the preferred digest to be used for PGP binary.

Standardwert:

```
$Self->{'PGP::Options::DigestPreference'} = 'sha256';
```

PGP::Key::Password

Sets the password for private PGP key.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PGP::Key::Password'} = {  
  '488A0B8F' => 'SomePassword',  
  'D2DF79FA' => 'SomePassword'  
};
```

PGP::TrustedNetwork

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.

Standardwert:

```
$Self->{'PGP::TrustedNetwork'} = '0';
```

PGP::Log

Configure your own log text for PGP.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$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.'  
};
```

PGP::StoreDecryptedData

If this option is enabled, then the decrypted data will be stored in the database if they are displayed in AgentTicketZoom.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PGP::StoreDecryptedData'} = '1';
```

PGP::Method

Sets the method PGP will use to sign and encrypt emails. Note Inline method is not compatible with RichText messages.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PGP::Method'} = 'Detached';
```

Framework → Crypt::SMIME

SMIME

Enables S/MIME support.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SMIME'} = '0';
```

SMIME::Bin

Defines the path to open ssl binary. It may need a HOME env (`$ENV{HOME} = '/var/lib/wwrun';`).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SMIME::Bin'} = '/usr/bin/openssl';
```

SMIME::CertPath

Specifies the directory where SSL certificates are stored.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SMIME::CertPath'} = '/etc/ssl/certs';
```

SMIME::PrivatePath

Specifies the directory where private SSL certificates are stored.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SMIME::PrivatePath'} = '/etc/ssl/private';
```

SMIME::CacheTTL

Cache time in seconds for the SSL certificate attributes.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SMIME::CacheTTL'} = '86400';
```

SMIME::StoreDecryptedData

If this option is enabled, then the decrypted data will be stored in the database if they are displayed in AgentTicketZoom.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SMIME::StoreDecryptedData'} = '1';
```

SMIME::FetchFromCustomer

Enables fetch S/MIME from CustomerUser backend support.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SMIME::FetchFromCustomer'} = '0';
```

Framework → CustomerInformationCenter

AgentCustomerInformationCenter::MainMenu###010-EditCustomerID

Main menu registration.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'AgentCustomerInformationCenter::MainMenu'}->{'010-EditCustomerID'} = {  
  'Link' => "[% Env('Baselink')  
%]Action=AdminCustomerCompany;Subaction=Change;CustomerID=[% Data.CustomerID | uri  
%];Nav=0",  
  'Name' => 'Edit customer company'  
};
```

Framework → Frontend::Admin

Events###Package

List of all Package events to be displayed in the GUI.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Events'}->{'Package'} = [  
  'PackageInstall',  
  'PackageReinstall',  
  'PackageUpgrade',  
  'PackageUninstall'  
];
```

Events###DynamicField

List of all DynamicField events to be displayed in the GUI.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Events'}->{'DynamicField'} = [  
  'DynamicFieldAdd',  
  'DynamicFieldUpdate',  
  'DynamicFieldDelete'  
];
```

Events###CustomerUser

List of all CustomerUser events to be displayed in the GUI.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Events'}->{'CustomerUser'} = [  
  'CustomerUserAdd',  
  'CustomerUserUpdate'  
];
```

Events###CustomerCompany

List of all CustomerCompany events to be displayed in the GUI.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Events'}->{'CustomerCompany'} = [
  'CustomerCompanyAdd',
  'CustomerCompanyUpdate'
];
```

Framework → Frontend::Admin::AdminCustomerCompany

AdminCustomerCompany::RunInitialWildcardSearch

Runs an initial wildcard search of the existing customer company when accessing the AdminCustomerCompany module.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'AdminCustomerCompany::RunInitialWildcardSearch'} = '1';
```

Framework → Frontend::Admin::AdminCustomerUser

AdminCustomerUser::RunInitialWildcardSearch

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'AdminCustomerUser::RunInitialWildcardSearch'} = '1';
```

Framework → Frontend::Admin::AdminSelectBox

AdminSelectBox::AllowDatabaseModification

Controls if the admin is allowed to make changes to the database via AdminSelectBox.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'AdminSelectBox::AllowDatabaseModification'} = '0';
```

Framework → Frontend::Admin::ModuleRegistration

Frontend::Module###Admin

Frontend module registration for the agent interface.

Standardwert:

```
$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::NavBar::ModuleAdmin'
},
'NavBarName' => 'Admin',
'Title' => ''
};

```

Frontend::Module###AdminInit

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AdminInit'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'NavBarName' => '',
  'Title' => 'Init'
};

```

Frontend::Module###AdminUser

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AdminUser'} = {
  'Description' => 'Create and manage agents.',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'Agent',
    'Description' => 'Create and manage agents.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Agents',
    'Prio' => '100'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Agents'
};

```

Frontend::Module###AdminGroup

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AdminGroup'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.Group.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Agent',
    'Description' => 'Create and manage groups.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Groups',
    'Prio' => '150'
  },
};

```

```
'NavBarName' => 'Admin',
'Title' => 'Groups'
};
```

Frontend::Module###AdminUserGroup

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminUserGroup'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.UserGroup.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Agent',
    'Description' => 'Link agents to groups.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Agents ↔ Groups',
    'Prio' => '200'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Agents ↔ Groups'
};
```

Frontend::Module###AdminCustomerUser

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminCustomerUser'} = {
  'Description' => 'Edit Customer Users.',
  'Group' => [
    'admin',
    'users'
  ],
  'GroupRo' => [
    ''
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketAction.js',
      'Core.Agent.Admin.CustomerUser.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => '',
      'Block' => 'ItemArea',
      'Description' => '',
      'Link' => 'Action=AdminCustomerUser;Nav=Agent',
      'LinkOption' => '',
      'Name' => 'Customer User Administration',
      'NavBar' => 'Customers',
      'Prio' => '9000',
      'Type' => ''
    }
  ],
  'NavBarModule' => {
    'Block' => 'Customer',
    'Description' => 'Create and manage customer users.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Customer User',
    'Prio' => '300'
  },
  'NavBarName' => 'Customers',
};
```

```
'Title' => 'Customer Users'
};
```

Frontend::Module###AdminCustomerCompany

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminCustomerCompany'} = {
  'Description' => 'Edit Customer Companies.',
  'Group' => [
    'admin',
    'users'
  ],
  'GroupRo' => [
    ''
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.CustomerCompany.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => '',
      'Block' => 'ItemArea',
      'Description' => '',
      'Link' => 'Action=AdminCustomerCompany;Nav=Agent',
      'LinkOption' => '',
      'Name' => 'Customer Administration',
      'NavBar' => 'Customers',
      'Prio' => '9100',
      'Type' => ''
    }
  ],
  'NavBarModule' => {
    'Block' => 'Customer',
    'Description' => 'Create and manage customers.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Customers',
    'Prio' => '310'
  },
  'NavBarName' => 'Customers',
  'Title' => 'Customer Companies'
};
```

Frontend::Module###AdminCustomerUserGroup

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminCustomerUserGroup'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.CustomerUserGroup.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Customer',
    'Description' => 'Link customer user to groups.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Customer User ↔ Groups',
    'Prio' => '400'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Customers ↔ Groups'
};
```

Frontend::Module###AdminCustomerUserService

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminCustomerUserService'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.CustomerUserService.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Customer',
    'Description' => 'Link customer user to services.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Customer User ↔ Services',
    'Prio' => '500'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Customer User ↔ Services'
};
```

Frontend::Module###AdminRole

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminRole'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.Role.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Agent',
    'Description' => 'Create and manage roles.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Roles',
    'Prio' => '600'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Roles'
};
```

Frontend::Module###AdminRoleUser

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminRoleUser'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.RoleUser.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Agent',
    'Description' => 'Link agents to roles.',
  }
```

```
'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
'Name' => 'Agents ↔ Roles',
'Prio' => '700'
},
'NavBarName' => 'Admin',
'Title' => 'Agents ↔ Roles'
};
```

Frontend::Module###AdminRoleGroup

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminRoleGroup'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.RoleGroup.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Agent',
    'Description' => 'Link roles to groups.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Roles ↔ Groups',
    'Prio' => '800'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Roles ↔ Groups'
};
```

Frontend::Module###AdminSMIME

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminSMIME'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.SMIME.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Email',
    'Description' => 'Manage S/MIME certificates for email encryption.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'S/MIME Certificates',
    'Prio' => '1100'
  },
  'NavBarName' => 'Admin',
  'Title' => 'S/MIME Management'
};
```

Frontend::Module###AdminPGP

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminPGP'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  ],
```

```
'NavBarModule' => {
  'Block' => 'Email',
  'Description' => 'Manage PGP keys for email encryption.',
  'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
  'Name' => 'PGP Keys',
  'Prio' => '1200'
},
'NavBarName' => 'Admin',
'Title' => 'PGP Key Management'
};
```

Frontend::Module###AdminMailAccount

Frontend module registration for the agent interface.

Standardwert:

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

Frontend::Module###AdminPostMasterFilter

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminPostMasterFilter'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.PostMasterFilter.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.PostMasterFilter.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Email',
    'Description' => 'Filter incoming emails.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'PostMaster Filters',
    'Prio' => '200'
  },
  'NavBarName' => 'Admin',
  'Title' => 'PostMaster Filters'
};
```

Frontend::Module###AdminEmail

Frontend module registration for the agent interface.

Standardwert:


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

Frontend::Module###AdminSession

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminSession'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.Session.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Manage existing sessions.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Session Management',
    'Prio' => '500'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Session Management'
};
```

Frontend::Module###AdminPerformanceLog

Frontend module registration for the agent interface.

Standardwert:

```
$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::NavBar::ModuleAdmin',
    'Name' => 'Performance Log',
    'Prio' => '550'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Performance Log'
};
```

Frontend::Module###AdminRegistration

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminRegistration'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.Registration.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.Registration.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Manage system registration.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'System Registration',
    'Prio' => '350'
  },
  'NavBarName' => 'Admin',
  'Title' => 'System Registration'
};
```

Frontend::Module###AdminOTRSBusiness

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminOTRSBusiness'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.OTRSBusiness.css'
    ]
  },
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Deploy and manage OTRS Business Solution™.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'OTRS Business Solution™',
    'Prio' => '360'
  },
  'NavBarName' => 'Admin',
  'Title' => 'OTRS Business Solution™'
};
```

Frontend::Module###AdminSupportDataCollector

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminSupportDataCollector'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.SupportDataCollector.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.SupportDataCollector.js'
    ]
  },
  'NavBarModule' => {
```

```
'Block' => 'System',
'Description' => 'Manage support data.',
'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
'Name' => 'Support Data Collector',
'Prio' => '370'
},
'NavBarName' => 'Admin',
'Title' => 'Support Data Collector'
};
```

Frontend::Module###AdminCloudServices

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminCloudServices'} = {
'Description' => 'Admin',
'Group' => [
'admin'
],
'Loader' => {
'CSS' => [
'Core.Agent.Admin.CloudServices.css'
]
},
'NavBarModule' => {
'Block' => 'System',
'Description' => 'Manage OTRS Group cloud services.',
'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
'Name' => 'Cloud Services',
'Prio' => '380'
},
'NavBarName' => 'Admin',
'Title' => 'Cloud Services'
};
```

Frontend::Module###AdminLog

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminLog'} = {
'Description' => 'Admin',
'Group' => [
'admin'
],
'Loader' => {
'JavaScript' => [
'Core.Agent.Admin.Log.js'
]
},
'NavBarModule' => {
'Block' => 'System',
'Description' => 'View system log messages.',
'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
'Name' => 'System Log',
'Prio' => '600'
},
'NavBarName' => 'Admin',
'Title' => 'System Log'
};
```

Frontend::Module###AdminSelectBox

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminSelectBox'} = {
'Description' => 'Admin',
'Group' => [
```

```

    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.SelectBox.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Execute SQL statements.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'SQL Box',
    'Prio' => '700'
  },
  'NavBarName' => 'Admin',
  'Title' => 'SQL Box'
};

```

Frontend::Module###AdminPackageManager

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AdminPackageManager'} = {
  'Description' => 'Software Package Manager.',
  'Group' => [
    'admin'
  ],
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Update and extend your system with software packages.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Package Manager',
    'Prio' => '1000'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Package Manager'
};

```

Frontend::Module###AdminSystemMaintenance

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AdminSystemMaintenance'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.SystemMaintenance.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Schedule a maintenance period.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'System Maintenance',
    'Prio' => '501'
  },
  'NavBarName' => 'Admin',
  'Title' => 'System Maintenance'
};

```

Frontend::Module###AdminCloudServiceSupportDataCollector

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AdminCloudServiceSupportDataCollector'} = {

```

```
'Description' => 'Admin',
'Group' => [
  'admin'
],
'Loader' => {
  'CSS' => [
    'Core.Agent.Admin.CloudService.SupportDataCollector.css'
  ]
},
'Title' => 'Support data collector'
};
```

Framework → Frontend::Agent

AgentLogo

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.

Standardwert:

```
$Self->{'AgentLogo'} = {
  'StyleHeight' => '85px',
  'StyleRight' => '38px',
  'StyleTop' => '4px',
  'StyleWidth' => '270px',
  'URL' => 'skins/Agent/default/img/logo_bg.png'
};
```

AgentLogoCustom###default

The logo shown in the header of the agent interface for the skin "default". See "Agent-Logo" for further description.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'AgentLogoCustom'}->{'default'} = {
  'StyleHeight' => '67px',
  'StyleRight' => '38px',
  'StyleTop' => '4px',
  'StyleWidth' => '270px',
  'URL' => 'skins/Agent/default/img/logo_bg.png'
};
```

AgentLogoCustom###slim

The logo shown in the header of the agent interface for the skin "slim". See "Agent-Logo" for further description.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'AgentLogoCustom'}->{'slim'} = {
  'StyleHeight' => '67px',
  'StyleRight' => '38px',
  'StyleTop' => '4px',
  'StyleWidth' => '270px',
  'URL' => 'skins/Agent/default/img/logo_bg.png'
};
```

AgentLogoCustom###ivory

The logo shown in the header of the agent interface for the skin "ivory". See "Agent-Logo" for further description.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'AgentLogoCustom'}->{'ivory'} = {
  'StyleHeight' => '67px',
  'StyleRight' => '38px',
  'StyleTop' => '4px',
  'StyleWidth' => '270px',
  'URL' => 'skins/Agent/default/img/logo_bg.png'
};
```

AgentLogoCustom###ivory-slim

The logo shown in the header of the agent interface for the skin "ivory-slim". See "AgentLogo" for further description.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'AgentLogoCustom'}->{'ivory-slim'} = {
  'StyleHeight' => '67px',
  'StyleRight' => '38px',
  'StyleTop' => '4px',
  'StyleWidth' => '270px',
  'URL' => 'skins/Agent/default/img/logo_bg.png'
};
```

AgentLoginLogo

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.

Standardwert:

```
$Self->{'AgentLoginLogo'} = {
  'StyleHeight' => '70px',
  'URL' => 'skins/Agent/default/img/loginlogo_default.png'
};
```

LoginURL

Defines an alternate URL, where the login link refers to.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'LoginURL'} = 'http://host.example.com/login.html';
```

LogoutURL

Defines an alternate URL, where the logout link refers to.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

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

PreApplicationModule###AgentInfo

Defines a useful module to load specific user options or to display news.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PreApplicationModule'}->{'AgentInfo'} = 'Kernel::Modules::AgentInfo';
```

InfoKey

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'InfoKey'} = 'wpt22';
```

InfoFile

File that is displayed in the Kernel::Modules::AgentInfo module, if located under Kernel/Output/HTML/Templates/Standard/AgentInfo.tt.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'InfoFile'} = 'AgentInfo';
```

LostPassword

Activates lost password feature for agents, in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'LostPassword'} = '1';
```

ShowMotd

Shows the message of the day on login screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'ShowMotd'} = '0';
```

NotificationSubjectLostPasswordToken

Defines the subject for notification mails sent to agents, with token about new requested password.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'NotificationSubjectLostPasswordToken'} = 'New OTRS password request';
```

NotificationBodyLostPasswordToken

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).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

NotificationSubjectLostPassword

Defines the subject for notification mails sent to agents, about new password.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'NotificationSubjectLostPassword'} = 'New OTRS password';
```

NotificationBodyLostPassword

Defines the body text for notification mails sent to agents, about new password (after using this link the new password will be sent).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$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_ContentType>://<OTRS_CONFIG_FQDN>/<OTRS_CONFIG_ScriptAlias>index.pl  
';
```

OpenMainMenuOnHover

If enabled, the first level of the main menu opens on mouse hover (instead of click only).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'OpenMainMenuOnHover'} = '0';
```

FirstnameLastnameOrder

Specifies the order in which the firstname and the lastname of agents will be displayed.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'FirstnameLastnameOrder'} = '0';
```

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

Default skin for the agent interface.

Standardwert:

```
$Self->{'Loader::Agent::Skin'}->{'000-default'} = {  
  'Description' => 'This is the default orange - black skin.',  
  'HomePage' => 'www.otrs.org',  
  'InternalName' => 'default',  
  'VisibleName' => 'Default'  
};
```

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

Default skin for the agent interface (slim version).

Standardwert:

```
$Self->{'Loader::Agent::Skin'}->{'001-slim'} = {  
  'Description' => '"Slim" skin which tries to save screen space for power users.',  
  'HomePage' => 'www.otrs.org',  
  'InternalName' => 'slim',  
  'VisibleName' => 'Default (Slim)'  
};
```


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

Balanced white skin by Felix Niklas.

Standardwert:

```
$Self->{'Loader::Agent::Skin'}->{'001-ivory'} = {  
  'Description' => 'Balanced white skin by Felix Niklas.',  
  'HomePage' => 'www.felixniklas.de',  
  'InternalName' => 'ivory',  
  'VisibleName' => 'Ivory'  
};
```

Loader::Agent::Skin###001-ivory-slim

Balanced white skin by Felix Niklas (slim version).

Standardwert:

```
$Self->{'Loader::Agent::Skin'}->{'001-ivory-slim'} = {  
  'Description' => 'Balanced white skin by Felix Niklas (slim version).',  
  'HomePage' => 'www.felixniklas.de',  
  'InternalName' => 'ivory-slim',  
  'VisibleName' => 'Ivory (Slim)'  
};
```

Loader::Agent::DefaultSelectedSkin

The agent skin's InternalName which should be used in the agent interface. Please check the available skins in Frontend::Agent::Skins.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Loader::Agent::DefaultSelectedSkin'} = 'default';
```

Loader::Agent::DefaultSelectedSkin::HostBased

It is possible to configure different skins, for example to distinguish between different 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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Loader::Agent::DefaultSelectedSkin::HostBased'} = {  
  'host1\\.example\\.com' => 'SomeSkin1',  
  'host2\\.example\\.com' => 'SomeSkin2'  
};
```

AutoComplete::Agent###Default

Defines the config options for the autocompletion feature.

Standardwert:

```
$Self->{'AutoComplete::Agent'}->{'Default'} = {  
  'AutoCompleteActive' => '1',  
  'ButtonText' => 'Search',  
  'MaxResultsDisplayed' => '20',  
  'MinQueryLength' => '2',  
  'QueryDelay' => '100'  
};
```

AutoComplete::Agent###CustomerSearch

Defines the config options for the autocompletion feature.

Standardwert:

```
$Self->{'AutoComplete::Agent'}->{'CustomerSearch'} = {
  'AutoCompleteActive' => '1',
  'ButtonText' => 'Search Customer',
  'MaxResultsDisplayed' => '20',
  'MinQueryLength' => '2',
  'QueryDelay' => '100'
};
```

AutoComplete::Agent###UserSearch

Defines the config options for the autocomplete feature.

Standardwert:

```
$Self->{'AutoComplete::Agent'}->{'UserSearch'} = {
  'AutoCompleteActive' => '1',
  'ButtonText' => 'Search User',
  'MaxResultsDisplayed' => '20',
  'MinQueryLength' => '2',
  'QueryDelay' => '100'
};
```

PossibleNextActions

Defines the list of possible next actions on an error screen, a full path is required, then is possible to add external links if needed.

Standardwert:

```
$Self->{'PossibleNextActions'} = {
  "[% Env(\ 'CGIHandle\ ' ) %]?Action=AgentDashboard" => 'Go to dashboard!'
};
```

ModernizeFormFields

Use new type of select and autocomplete fields in agent interface, where applicable (InputFields).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'ModernizeFormFields'} = '1';
```

Framework → Frontend::Agent::Auth::TwoFactor

AuthTwoFactorModule

Defines the two-factor module to authenticate agents.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'AuthTwoFactorModule'} =
  'Kernel::System::Auth::TwoFactor::GoogleAuthenticator';
```

AuthTwoFactorModule::SecretPreferencesKey

Defines the agent preferences key where the shared secret key is stored.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'AuthTwoFactorModule::SecretPreferencesKey'} =
  'UserGoogleAuthenticatorSecretKey';
```

AuthTwoFactorModule::AllowEmptySecret

Defines if agents should be allowed to login if they have no shared secret stored in their preferences and therefore are not using two-factor authentication.

Standardwert:

```
$Self->{'AuthTwoFactorModule::AllowEmptySecret'} = '1';
```

AuthTwoFactorModule::AllowPreviousToken

Defines if the previously valid token should be accepted for authentication. This is slightly less secure but gives users 30 seconds more time to enter their one-time password.

Standardwert:

```
$Self->{'AuthTwoFactorModule::AllowPreviousToken'} = '1';
```

Framework → Frontend::Agent::Dashboard

AgentCustomerInformationCenter::Backend###0600-CIC-CustomerCompanyInformation

Parameters for the dashboard backend of the customer company information of the agent interface. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.

Standardwert:

```
$Self->{'AgentCustomerInformationCenter::Backend'}->{'0600-CIC-CustomerCompanyInformation'} = {
  'Attributes' => '',
  'Block' => 'ContentSmall',
  'Default' => '1',
  'Description' => 'Customer Information',
  'Group' => '',
  'Module' => 'Kernel::Output::HTML::Dashboard::CustomerCompanyInformation',
  'Title' => 'Customer Information'
};
```

DashboardBackend###0000-ProductNotify

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.

Standardwert:

```
$Self->{'DashboardBackend'}->{'0000-ProductNotify'} = {
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '1440',
  'Default' => '1',
  'Description' => 'News about OTRS releases!',
  'Group' => 'admin',
  'Module' => 'Kernel::Output::HTML::Dashboard::ProductNotify',
  'Title' => 'Product News'
};
```

DashboardBackend###0390-UserOutOfOffice

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.

Standardwert:

```
$Self->{'DashboardBackend'}->{'0390-UserOutOfOffice'} = {
  'Block' => 'ContentSmall',
```

```
'CacheTTLLocal' => '5',
'Default' => '1',
'Description' => '',
'Group' => '',
'IdleMinutes' => '60',
'Limit' => '10',
'Module' => 'Kernel::Output::HTML::Dashboard::UserOutOfOffice',
'SortBy' => 'UserFullname',
'Title' => 'Out Of Office'
};
```

DashboardBackend###0400-UserOnline

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.

Standardwert:

```
$Self->{'DashboardBackend'}->{'0400-UserOnline'} = {
  'Block' => 'ContentSmall',
  'CacheTTLLocal' => '5',
  'Default' => '0',
  'Description' => '',
  'Filter' => 'Agent',
  'Group' => '',
  'IdleMinutes' => '60',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::Dashboard::UserOnline',
  'ShowEmail' => '0',
  'SortBy' => 'UserFullname',
  'Title' => 'OnLine'
};
```

DashboardBackend###0405-News

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.

Standardwert:

```
$Self->{'DashboardBackend'}->{'0405-News'} = {
  'Block' => 'ContentSmall',
  'CacheTTL' => '360',
  'Default' => '1',
  'Description' => '',
  'Group' => '',
  'Limit' => '6',
  'Module' => 'Kernel::Output::HTML::Dashboard::News',
  'Title' => 'OTRS News'
};
```

DashboardBackend###0410-RSS

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'DashboardBackend'}->{'0410-RSS'} = {
```

```
'Block' => 'ContentSmall',
'CacheTTL' => '360',
'Default' => '1',
'Description' => '',
'Group' => '',
'Limit' => '6',
'Module' => 'Kernel::Output::HTML::Dashboard::RSS',
'Title' => 'Custom RSS Feed',
'URL' => 'http://www.otrs.com/en/rss.xml',
'URL_de' => 'http://www.otrs.com/de/rss.xml',
'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'
};
```

DashboardBackend###0420-CommandOutput

Defines the parameters for the dashboard backend. "Cmd" is used to specify command with parameters. "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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'DashboardBackend'}->{'0420-CommandOutput'} = {
'Block' => 'ContentSmall',
'CacheTTL' => '60',
'Cmd' => '/bin/echo Configure me please.',
'Default' => '0',
'Description' => '',
'Group' => '',
'Module' => 'Kernel::Output::HTML::Dashboard::CommandOutput',
'Title' => 'Sample command output'
};
```

DashboardBackend###0200-Image

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$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::Dashboard::Image',
'Title' => 'A picture',
'URL' => 'http://www.otrs.com/wp-uploads//2013/10/OTRS_Logo-300x170.png',
'Width' => '198'
};
```

DashboardBackend###0210-MOTD

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'DashboardBackend'}->{'0210-MOTD'} = {
  'Block' => 'ContentLarge',
  'Default' => '1',
  'Group' => '',
  'Module' => 'Kernel::Output::HTML::Dashboard::MOTD',
  'Title' => 'Message of the Day'
};
```

DashboardBackend###0300-IFrame

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$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::Dashboard::IFrame',
  'Scrolling' => 'auto',
  'Title' => 'A Website',
  'URL' => 'http://www.otrs.org/',
  'Width' => '1024'
};
```

AgentCustomerInformationCenter::Backend###0050-CIC-CustomerUserList

Parameters for the dashboard backend of the customer user list overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTL-Local" is the cache time in minutes for the plugin.

Standardwert:

```
$Self->{'AgentCustomerInformationCenter::Backend'}->{'0050-CIC-CustomerUserList'} = {
  'Attributes' => '',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'Description' => 'All customer users of a CustomerID',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::Dashboard::CustomerUserList',
  'Permission' => 'ro',
  'Title' => 'Customer Users'
};
```

Framework → Frontend::Agent::LinkObject

Frontend::AgentLinkObject::WildcardSearch

Starts a wildcard search of the active object after the link object mask is started.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::AgentLinkObject::WildcardSearch'} = '0';
```

Framework → Frontend::Agent::ModuleMetaHead

Frontend::HeaderMetaModule###100-Refresh

Defines the module to generate code for periodic page reloads.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::HeaderMetaModule'}->{'100-Refresh'} = {  
  'Module' => 'Kernel::Output::HTML::HeaderMeta::Refresh'  
};
```

Framework → Frontend::Agent::ModuleNotify

Frontend::NotifyModule###1100-OTRSBusiness

Defines the module to display a notification in different interfaces on different occasions for OTRS Business Solution™.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::NotifyModule'}->{'1100-OTRSBusiness'} = {  
  'Group' => 'admin',  
  'Module' => 'Kernel::Output::HTML::Notification::AgentOTRSBusiness'  
};
```

Frontend::NotifyModule###2000-UID-Check

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).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::NotifyModule'}->{'2000-UID-Check'} = {  
  'Module' => 'Kernel::Output::HTML::Notification::UIDCheck'  
};
```

Frontend::NotifyModule###2500-AgentSessionLimit

Defines the module to display a notification in the agent interface, if the agent session limit prior warning is reached.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::NotifyModule'}->{'2500-AgentSessionLimit'} = {  
  'Module' => 'Kernel::Output::HTML::Notification::AgentSessionLimit'  
};
```

Frontend::NotifyModule###3000-ShowAgentOnline

Defines the module that shows all the currently logged in agents in the agent interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::NotifyModule'}->{'3000-ShowAgentOnline'} = {
```

```
'IdleMinutes' => '60',
'Module' => 'Kernel::Output::HTML::Notification::AgentOnline',
'ShowEmail' => '1'
};
```

Frontend::NotifyModule###4000-ShowCustomerOnline

Defines the module that shows all the currently logged in customers in the agent interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::NotifyModule'}->{'4000-ShowCustomerOnline'} = {
  'IdleMinutes' => '60',
  'Module' => 'Kernel::Output::HTML::Notification::CustomerOnline',
  'ShowEmail' => '1'
};
```

Frontend::NotifyModule###5500-OutofOffice-Check

Defines the module to display a notification in the agent interface, if the agent is logged in while having out-of-office active.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::NotifyModule'}->{'5500-OutofOffice-Check'} = {
  'Module' => 'Kernel::Output::HTML::Notification::OutofOfficeCheck'
};
```

Frontend::NotifyModule###6000-SystemMaintenance-Check

Defines the module to display a notification in the agent interface, if the agent is logged in while having system maintenance active.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::NotifyModule'}->{'6000-SystemMaintenance-Check'} = {
  'Module' => 'Kernel::Output::HTML::Notification::SystemMaintenanceCheck'
};
```

Frontend::NotifyModule###7000-AgentTimeZone-Check

Defines the module to display a notification in the agent interface, if the agent has not yet selected a time zone.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::NotifyModule'}->{'7000-AgentTimeZone-Check'} = {
  'Module' => 'Kernel::Output::HTML::Notification::AgentTimeZoneCheck'
};
```

Frontend::NotifyModule###9000-Generic

Defines the module that shows a generic notification in the agent interface. Either "Text" - if configured - or the contents of "File" will be displayed.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::NotifyModule'}->{'9000-Generic'} = {
  'File' => '<OTRS_CONFIG_Home>/var/notify.txt',
  'Link' => 'http://www.otrs.com',
};
```



```
'Module' => 'Kernel::Output::HTML::Notification::Generic',
'Priority' => 'Warning',
'Text' => 'The OTRS Website'
};
```

Framework → Frontend::Agent::ModuleRegistration

Frontend::Module###Logout

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'Logout'} = {
  'Description' => 'Logout',
  'NavBarName' => '',
  'Title' => ''
};
```

Frontend::Module###AgentDashboard

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentDashboard'} = {
  'Description' => 'Agent Dashboard',
  'Loader' => {
    'CSS' => [
      'Core.Agent.Dashboard.css',
      'Core.AllocationList.css',
      'thirdparty/fullcalendar-2.8.0/fullcalendar.min.css',
      'thirdparty/nvd3-1.7.1/nv.d3.css'
    ],
    'JavaScript' => [
      'thirdparty/momentjs-2.13.0/moment.min.js',
      'thirdparty/fullcalendar-2.8.0/fullcalendar.min.js',
      'thirdparty/d3-3.5.6/d3.min.js',
      'thirdparty/nvd3-1.7.1/nvd3.min.js',
      'thirdparty/nvd3-1.7.1/models/OTRSLineChart.js',
      'thirdparty/nvd3-1.7.1/models/OTRSMultiBarChart.js',
      'thirdparty/nvd3-1.7.1/models/OTRSStackedAreaChart.js',
      'thirdparty/canvg-1.4/rgbcolor.js',
      'thirdparty/canvg-1.4/StackBlur.js',
      'thirdparty/canvg-1.4/canvg.js',
      'thirdparty/StringView-8/stringview.js',
      'Core.UI.AdvancedChart.js',
      'Core.UI.AllocationList.js',
      'Core.Agent.TableFilters.js',
      'Core.Agent.Dashboard.js',
      'Core.Agent.Statistics.ParamsWidget.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'd',
      'Block' => 'ItemArea',
      'Description' => '',
      'Link' => 'Action=AgentDashboard',
      'LinkOption' => '',
      'Name' => 'Dashboard',
      'NavBar' => 'Dashboard',
      'Prio' => '50',
      'Type' => 'Menu'
    }
  ],
  'NavBarName' => 'Dashboard',
  'Title' => ''
};
```

Frontend::Module###AgentCustomerInformationCenter

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentCustomerInformationCenter'} = {
  'Description' => 'Customer Information Center.',
  'Loader' => {
    'CSS' => [
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.UI.AllocationList.js',
      'Core.Agent.Dashboard.js',
      'Core.Agent.TableFilters.js',
      'Core.Agent.CustomerInformationCenter.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'c',
      'Block' => 'ItemArea',
      'Description' => '',
      'Link' => 'Action=AgentCustomerInformationCenter',
      'LinkOption' => 'onclick="window.setTimeout(function()
(Core.Agent.CustomerInformationCenterSearch.OpenSearchDialog());, 0); return false;"',
      'Name' => 'Customer Information Center',
      'NavBar' => 'Customers',
      'Prio' => '50',
      'Type' => ''
    },
    {
      'AccessKey' => '',
      'Block' => 'ItemArea',
      'Description' => '',
      'Link' => 'Action=AgentCustomerInformationCenter',
      'LinkOption' => '',
      'Name' => 'Customers',
      'NavBar' => 'Customers',
      'Prio' => '60',
      'Type' => 'Menu'
    }
  ],
  'NavBarName' => 'Customers',
  'Title' => ''
};
```

Frontend::Module###AgentCustomerInformationCenterSearch

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentCustomerInformationCenterSearch'} = {
  'Description' => 'Customer Information Center Search.',
  'Title' => ''
};
```

Frontend::Module###AgentPreferences

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentPreferences'} = {
  'Description' => 'Agent Preferences.',
  'Loader' => {
    'CSS' => [
      'Core.Agent.Preferences.css'
    ],
    'JavaScript' => [
      'Core.Agent.Preferences.js'
    ]
  },
  'NavBarName' => 'Preferences',
  'Title' => ''
};
```

```
};
```

Frontend::Module###PictureUpload

Frontend module registration for the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::Module'}->{'PictureUpload'} = {  
  'Description' => 'Picture upload module.',  
  'NavBarName' => 'Ticket',  
  'Title' => 'Picture Upload'  
};
```

Frontend::Module###AgentSpelling

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentSpelling'} = {  
  'Description' => 'Spell checker.',  
  'Loader' => {  
    'JavaScript' => [  
      'Core.Agent.TicketAction.js'  
    ]  
  },  
  'NavBarName' => '',  
  'Title' => 'Spell Checker'  
};
```

Frontend::Module###SpellingInline

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'SpellingInline'} = {  
  'Description' => 'Spell checker.',  
  'NavBarName' => '',  
  'Title' => 'Spell Checker'  
};
```

Frontend::Module###AgentBook

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentBook'} = {  
  'Description' => 'Address book of CustomerUser sources.',  
  'Loader' => {  
    'JavaScript' => [  
      'Core.Agent.CustomerSearch.js',  
      'Core.Agent.CustomerSearchAutoComplete.js',  
      'Core.Agent.TicketAction.js',  
      'Core.Agent.Book.js'  
    ]  
  },  
  'NavBarName' => '',  
  'Title' => 'Address Book'  
};
```

Frontend::Module###AgentLinkObject

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentLinkObject'} = {  
  'Description' => 'Link Object.',  
};
```

```
'Loader' => {
  'JavaScript' => [
    'Core.Agent.LinkObject.SearchForm.js'
  ]
},
'NavBarName' => '',
'Title' => 'Link Object'
};
```

Frontend::Module###AgentInfo

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentInfo'} = {
  'Description' => 'Generic Info module.',
  'NavBarName' => '',
  'Title' => 'Info'
};
```

Frontend::Module###AgentSearch

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentSearch'} = {
  'Description' => 'Global Search Module.',
  'NavBarName' => '',
  'Title' => 'Search'
};
```

Frontend::Module###AgentOTRSBusiness

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentOTRSBusiness'} = {
  'Description' => 'Agent',
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.OTRSBusiness.css'
    ]
  },
  'NavBarName' => '',
  'Title' => 'OTRS Business Solution™'
};
```

CustomerFrontend::Module###SpellingInline

Frontend module registration for the customer interface.

Standardwert:

```
$Self->{'CustomerFrontend::Module'}->{'SpellingInline'} = {
  'Description' => 'Spell checker.',
  'NavBarName' => '',
  'Title' => 'Spell Checker'
};
```

Frontend::Module###AgentHTMLReference

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentHTMLReference'} = {
  'Description' => 'HTML Reference.',
  'Group' => [
    'users'
  ],
};
```

```
'GroupRo' => [
  'users'
],
'Loader' => {
  'CSS' => [
    'Core.Agent.HTMLReference.css'
  ]
},
'NavBarName' => '',
'Title' => 'HTML Reference'
};
```

Frontend::Module###AgentStatistics

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentStatistics'} = {
  'Description' => '',
  'Group' => [
    'stats'
  ],
  'GroupRo' => [
    'stats'
  ],
  'Loader' => {
    'CSS' => [
      'thirdparty/nvd3-1.7.1/nv.d3.css',
      'Core.Agent.Statistics.css'
    ],
    'JavaScript' => [
      'thirdparty/d3-3.5.6/d3.min.js',
      'thirdparty/nvd3-1.7.1/nvd3.min.js',
      'thirdparty/nvd3-1.7.1/models/OTRSLineChart.js',
      'thirdparty/nvd3-1.7.1/models/OTRSMultiBarChart.js',
      'thirdparty/nvd3-1.7.1/models/OTRSStackedAreaChart.js',
      'thirdparty/canvg-1.4/rgbcolor.js',
      'thirdparty/canvg-1.4/StackBlur.js',
      'thirdparty/canvg-1.4/canvg.js',
      'thirdparty/StringView-8/stringview.js',
      'Core.Agent.Statistics.js',
      'Core.UI.AdvancedChart.js',
      'Core.Agent.Statistics.ParamsWidget.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => '',
      'Block' => 'ItemArea',
      'Description' => '',
      'Link' => 'Action=AgentStatistics;Subaction=Overview',
      'LinkOption' => '',
      'Name' => 'Reports',
      'NavBar' => 'Reports',
      'Prio' => '8500',
      'Type' => 'Menu'
    },
    {
      'AccessKey' => '',
      'Block' => '',
      'Description' => '',
      'GroupRo' => [
        'stats'
      ],
      'Link' => 'Action=AgentStatisticsReports;Subaction=Overview',
      'LinkOption' => 'class="OTRSBusinessRequired"',
      'Name' => 'Reports (OTRS Business Solution™)',
      'NavBar' => 'Reports',
      'Prio' => '100',
      'Type' => ''
    }
  ],
}
```

```

'AccessKey' => '',
'Block' => '',
'Description' => '',
'GroupRo' => [
  'stats'
],
'Link' => 'Action=AgentStatistics;Subaction=0verview',
'LinkOption' => '',
'Name' => 'Statistics',
'NavBar' => 'Reports',
'Prio' => '200',
'Type' => ''
}
],
'NavBarName' => 'Reports',
'Title' => 'Statistics'
};

```

Framework → Frontend::Agent::NavBarModule

Frontend::NavBarModule###6-CustomerCompany

Frontend module registration (disable company link if no company feature is used).

Standardwert:

```

$self->{'Frontend::NavBarModule'}->{'6-CustomerCompany'} = {
  'Module' => 'Kernel::Output::HTML::NavBar::CustomerCompany'
};

```

Frontend::NavBarModule###7-AgentTicketService

Frontend module registration (disable AgentTicketService link if Ticket Service feature is not used).

Standardwert:

```

$self->{'Frontend::NavBarModule'}->{'7-AgentTicketService'} = {
  'Module' => 'Kernel::Output::HTML::NavBar::AgentTicketService'
};

```

Framework → Frontend::Agent::Preferences

PreferencesTable

Defines the name of the table where the user preferences are stored.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```

$self->{'PreferencesTable'} = 'user_preferences';

```

PreferencesTableKey

Defines the column to store the keys for the preferences table.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```

$self->{'PreferencesTableKey'} = 'preferences_key';

```

PreferencesTableValue

Defines the name of the column to store the data in the preferences table.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PreferencesTableValue'} = 'preferences_value';
```

PreferencesTableUserID

Defines the name of the column to store the user identifier in the preferences table.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PreferencesTableUserID'} = 'user_id';
```

PreferencesView

Sets the display order of the different items in the preferences view.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PreferencesView'} = [
  'User Profile',
  'Notification Settings',
  'Other Settings'
];
```

PreferencesGroups###Password

Defines the config parameters of this item, to be shown in the preferences view. 'PasswordRegExp' allows to match passwords against a regular expression. Define the minimum number of characters using 'PasswordMinSize'. Define if at least 2 lowercase and 2 uppercase letter characters are needed by setting the appropriate option to '1'. 'PasswordMin2Characters' defines if the password needs to contain at least 2 letter characters (set to 0 or 1). 'PasswordNeedDigit' controls the need of at least 1 digit (set to 0 or 1 to control). 'PasswordMaxLoginFailed' allows to set an agent to invalid-temporarily if max failed logins reached.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'Password'} = {
  'Active' => '1',
  'Area' => 'Agent',
  'Column' => 'User Profile',
  'Label' => 'Change password',
  'Module' => 'Kernel::Output::HTML::Preferences::Password',
  'PasswordMaxLoginFailed' => '0',
  'PasswordMin2Characters' => '0',
  'PasswordMin2Lower2UpperCharacters' => '0',
  'PasswordMinSize' => '0',
  'PasswordNeedDigit' => '0',
  'PasswordRegExp' => '',
  'Prio' => '0500'
};
```

PreferencesGroups###GoogleAuthenticatorSecretKey

Defines the config parameters of this item, to be shown in the preferences view.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'GoogleAuthenticatorSecretKey'} = {
  'Active' => '0',
  'Block' => 'Input',
  'Column' => 'User Profile',
  'Desc' => 'Enter your shared secret to enable two factor authentication.',
  'Key' => 'Shared Secret',
  'Label' => 'Google Authenticator',
  'Module' => 'Kernel::Output::HTML::Preferences::Generic',
  'PrefKey' => 'UserGoogleAuthenticatorSecretKey',
  'Prio' => '0600'
};
```

PreferencesGroups###SpellDict

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$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::Preferences::Generic',  
  'PrefKey' => 'UserSpellDict',  
  'Prio' => '2000'  
};
```

PreferencesGroups###Comment

Defines the config parameters of this item, to be shown in the preferences view.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'Comment'} = {  
  'Active' => '0',  
  'Block' => 'Input',  
  'Column' => 'Other Settings',  
  'Data' => "[% Env('UserComment') %]",  
  'Key' => 'Comment',  
  'Label' => 'Comment',  
  'Module' => 'Kernel::Output::HTML::Preferences::Generic',  
  'PrefKey' => 'UserComment',  
  'Prio' => '6000'  
};
```

PreferencesGroups###Language

Defines the config parameters of this item, to be shown in the preferences view.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'Language'} = {  
  'Active' => '1',  
  'Column' => 'User Profile',  
  'Key' => 'Language',  
  'Label' => 'Language',  
  'Module' => 'Kernel::Output::HTML::Preferences::Language',  
  'PrefKey' => 'UserLanguage',  
  'Prio' => '1000'  
};
```

PreferencesGroups###Skin

Defines the config parameters of this item, to be shown in the preferences view.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'Skin'} = {  
  'Active' => '1',  
  'Column' => 'Other Settings',  
  'Key' => 'Skin',  
  'Label' => 'Skin',  
  'Module' => 'Kernel::Output::HTML::Preferences::Skin',  
  'PrefKey' => 'UserSkin',  
  'Prio' => '100'  
};
```


PreferencesGroups###Theme

Defines the config parameters of this item, to be shown in the preferences view.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'Theme'} = {
  'Active' => '1',
  'Column' => 'User Profile',
  'Key' => 'Frontend theme',
  'Label' => 'Theme',
  'Module' => 'Kernel::Output::HTML::Preferences::Theme',
  'PrefKey' => 'UserTheme',
  'Prio' => '3000'
};
```

PreferencesGroups###TimeZone

Defines the config parameters of this item, to be shown in the preferences view.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'TimeZone'} = {
  'Active' => '1',
  'Column' => 'User Profile',
  'Key' => 'Time Zone',
  'Label' => 'Time Zone',
  'Module' => 'Kernel::Output::HTML::Preferences::TimeZone',
  'PrefKey' => 'UserTimeZone',
  'Prio' => '3500'
};
```

PreferencesGroups###OutOfOffice

Defines the config parameters of this item, to be shown in the preferences view.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'OutOfOffice'} = {
  'Active' => '1',
  'Block' => 'OutOfOffice',
  'Column' => 'User Profile',
  'Key' => '',
  'Label' => 'Out Of Office Time',
  'Module' => 'Kernel::Output::HTML::Preferences::OutOfOffice',
  'PrefKey' => 'UserOutOfOffice',
  'Prio' => '4000'
};
```

PreferencesGroups###CSVSeparator

Gives end users the possibility to override the separator character for CSV files, defined in the translation files.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$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::Preferences::Generic',
'PrefKey' => 'UserCSVSeparator',
'Prio' => '4000'
};
```

Framework → Frontend::Agent::SearchRouter

Frontend::SearchDefault

Search backend default router.

Standardwert:

```
$Self->{'Frontend::SearchDefault'} = 'Action=AgentTicketSearch;Subaction=AJAX';
```

Framework → Frontend::Agent::Stats

Stats::SearchPageShown

Defines the default maximum number of statistics per page on the overview screen.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Stats::SearchPageShown'} = '50';
```

Stats::DefaultSelectedDynamicObject

Defines the default selection at the drop down menu for dynamic objects (Form: Common Specification).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Stats::DefaultSelectedDynamicObject'} = 'Ticket';
```

Stats::DefaultSelectedPermissions

Defines the default selection at the drop down menu for permissions (Form: Common Specification).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Stats::DefaultSelectedPermissions'} = [
  'stats'
];
```

Stats::DefaultSelectedFormat

Defines the default selection at the drop down menu for stats format (Form: Common Specification). Please insert the format key (see Stats::Format).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Stats::DefaultSelectedFormat'} = [
  'Print',
  'CSV',
  'Excel',
  'D3::BarChart',
  'D3::LineChart',
  'D3::StackedAreaChart'
];
```

Stats::SearchLimit

Defines the search limit for the stats.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Stats::SearchLimit'} = '1000';
```

Stats::Format

Defines all the possible stats output formats.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Stats::Format'} = {  
  'CSV' => 'CSV',  
  'D3::BarChart' => 'Graph: Bar Chart',  
  'D3::LineChart' => 'Graph: Line Chart',  
  'D3::StackedAreaChart' => 'Graph: Stacked Area Chart',  
  'Excel' => 'Excel',  
  'Print' => 'Print'  
};
```

Stats::ExchangeAxis

Allows agents to exchange the axis of a stat if they generate one.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Stats::ExchangeAxis'} = '0';
```

Stats::UseAgentElementInStats

Allows agents to generate individual-related stats.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Stats::UseAgentElementInStats'} = '0';
```

Stats::UseInvalidAgentInStats

Allows invalid agents to generate individual-related stats.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Stats::UseInvalidAgentInStats'} = '1';
```

Stats::CustomerIDAsMultiSelect

Shows all the customer identifiers in a multi-select field (not useful if you have a lot of customer identifiers).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Stats::CustomerIDAsMultiSelect'} = '1';
```

Framework → Frontend::Customer

CustomerHeadline

The headline shown in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerHeadline'} = 'Example Company';
```

CustomerLogo

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'CustomerLogo'} = {  
  'StyleHeight' => '50px',  
  'StyleRight' => '25px',  
  'StyleTop' => '2px',  
  'StyleWidth' => '135px',  
  'URL' => 'skins/Customer/default/img/logo.png'  
};
```

CustomerPanelUserID

Defines the user identifier for the customer panel.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerPanelUserID'} = '1';
```

CustomerGroupSupport

Activates support for customer groups.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerGroupSupport'} = '0';
```

CustomerGroupAlwaysGroups

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).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerGroupAlwaysGroups'} = [  
  'users'  
];
```

CustomerPanelLoginURL

Defines an alternate login URL for the customer panel..

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'CustomerPanelLoginURL'} = 'http://host.example.com/cgi-bin/login.pl';
```

CustomerPanelLogoutURL

Defines an alternate logout URL for the customer panel.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'CustomerPanelLogoutURL'} = 'http://host.example.com/cgi-bin/login.pl';
```

Frontend::CustomerUser::Item###1-GoogleMaps

Defines a customer item, which generates a google maps icon at the end of a customer info block.

Standardwert:

```
$Self->{'Frontend::CustomerUser::Item'}->{'1-GoogleMaps'} = {
  'Attributes' => 'UserStreet;UserCity;UserCountry;',
  'CSS' => 'Core.Agent.CustomerUser.GoogleMaps.css',
  'CSSClass' => 'GoogleMaps',
  'IconName' => 'fa-globe',
  'Module' => 'Kernel::Output::HTML::CustomerUser::Generic',
  'Required' => 'UserStreet;UserCity;',
  'Target' => '_blank',
  'Text' => 'Location',
  'URL' => 'http://maps.google.com/maps?z=7&q='
};
```

Frontend::CustomerUser::Item###2-Google

Defines a customer item, which generates a google icon at the end of a customer info block.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::CustomerUser::Item'}->{'2-Google'} = {
  'Attributes' => 'UserFirstname;UserLastname;',
  'CSS' => 'Core.Agent.CustomerUser.Google.css',
  'CSSClass' => 'Google',
  'IconName' => 'fa-google',
  'Module' => 'Kernel::Output::HTML::CustomerUser::Generic',
  'Required' => 'UserFirstname;UserLastname;',
  'Target' => '_blank',
  'Text' => 'Google',
  'URL' => 'http://google.com/search?q='
};
```

Frontend::CustomerUser::Item###2-LinkedIn

Defines a customer item, which generates a LinkedIn icon at the end of a customer info block.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::CustomerUser::Item'}->{'2-LinkedIn'} = {
  'Attributes' => 'UserFirstname;UserLastname;',
  'CSS' => 'Core.Agent.CustomerUser.Linkedin.css',
  'CSSClass' => 'LinkedIn',
  'IconName' => 'fa-linkedin',
  'Module' => 'Kernel::Output::HTML::CustomerUser::Generic',
  'Required' => 'UserFirstname;UserLastname;',
  'Target' => '_blank',
  'Text' => 'LinkedIn',
  'URL' => 'http://www.linkedin.com/commonSearch?type=people&keywords='
};
```

Frontend::CustomerUser::Item###3-XING

Defines a customer item, which generates a XING icon at the end of a customer info block.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::CustomerUser::Item'}->{'3-XING'} = {
  'Attributes' => 'UserFirstname;UserLastname;',
```

```
'CSS' => 'Core.Agent.CustomerUser.Xing.css',
'CSSClass' => 'Xing',
'IconName' => 'fa-xing',
'Module' => 'Kernel::Output::HTML::CustomerUser::Generic',
'Required' => 'UserFirstname;UserLastname;',
'Target' => '_blank',
'Text' => 'XING',
'URL' => 'https://www.xing.com/app/search?op=search;keywords='
};
```

CustomerPanelPreApplicationModule###CustomerAccept

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'CustomerPanelPreApplicationModule'}->{'CustomerAccept'} =
'Kernel::Modules::CustomerAccept';
```

CustomerPanel::InfoKey

Defines the key to check with CustomerAccept. If this user preferences key is true, then the message is accepted by the system.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'CustomerPanel::InfoKey'} = 'CustomerAccept1';
```

CustomerPanel::InfoFile

Defines the path of the shown info file, that is located under Kernel/Output/HTML/Templates/Standard/CustomerAccept.tt.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'CustomerPanel::InfoFile'} = 'CustomerAccept';
```

CustomerPanelLostPassword

Activates lost password feature for customers.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerPanelLostPassword'} = '1';
```

CustomerPanelCreateAccount

Enables customers to create their own accounts.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerPanelCreateAccount'} = '1';
```

CustomerPanelCreateAccount::MailRestrictions::Whitelist

If active, one of the regular expressions has to match the user's email address to allow registration.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'CustomerPanelCreateAccount::MailRestrictions::Whitelist'} = [
  '\\@your\\.domain\\.example$'
];
```

CustomerPanelCreateAccount::MailRestrictions::Blacklist

If active, none of the regular expressions may match the user's email address to allow registration.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'CustomerPanelCreateAccount::MailRestrictions::Blacklist'} = [
  '\\@your\\.domain\\.example$'
];
```

CustomerPanelSubjectLostPasswordToken

Defines the subject for notification mails sent to customers, with token about new requested password.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerPanelSubjectLostPasswordToken'} = 'New OTRS password request';
```

CustomerPanelBodyLostPasswordToken

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).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$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_ContentType>://<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.
';
```

CustomerPanelSubjectLostPassword

Defines the subject for notification mails sent to customers, about new password.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerPanelSubjectLostPassword'} = 'New OTRS password';
```

CustomerPanelBodyLostPassword

Defines the body text for notification mails sent to customers, about new password (after using this link the new password will be sent).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerPanelBodyLostPassword'} = 'Hi <OTRS_USERFIRSTNAME>,

New password: <OTRS_NEWPW>

<OTRS_CONFIG_ContentType>://<OTRS_CONFIG_FQDN>/<OTRS_CONFIG_ScriptAlias>customer.pl
';
```

CustomerPanelSubjectNewAccount

Defines the subject for notification mails sent to customers, about new account.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerPanelSubjectNewAccount'} = 'New OTRS Account!';
```

CustomerPanelBodyNewAccount

Defines the body text for notification mails sent to customers, about new account.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerPanelBodyNewAccount'} = 'Hi <OTRS_USERFIRSTNAME>,

You or someone impersonating you has created a new OTRS account for
you.

Full name: <OTRS_USERFIRSTNAME> <OTRS_USERLASTNAME>
User name: <OTRS_USERLOGIN>
Password : <OTRS_USERPASSWORD>

You can log in via the following URL. We encourage you to change your password
via the Preferences button after logging in.

<OTRS_CONFIG_ContentType>://<OTRS_CONFIG_FQDN>/<OTRS_CONFIG_ScriptAlias>customer.pl
';
```

Loader::Customer::Skin###000-default

Default skin for the customer interface.

Standardwert:

```
$Self->{'Loader::Customer::Skin'}->{'000-default'} = {
  'Description' => 'This is the default orange - black skin for the customer
interface.',
  'HomePage' => 'www.otrs.org',
  'InternalName' => 'default',
  'VisibleName' => 'Default'
};
```

Loader::Customer::SelectedSkin

The customer skin's InternalName which should be used in the customer interface. Please check the available skins in Frontend::Customer::Skins.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Loader::Customer::SelectedSkin'} = 'default';
```

Loader::Customer::SelectedSkin::HostBased

It is possible to configure different skins, for example to distinguish between different customers, to be used on a per-domain basis within the application. Using a regular expression (regex), you can configure a Key/Content pair to match a domain. The value in "Key" should match the domain, and the value in "Content" should be a valid skin on your system. Please see the example entries for the proper form of the regex.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Loader::Customer::SelectedSkin::HostBased'} = {  
  'host1\\.example\\.com' => 'Someskin1',  
  'host2\\.example\\.com' => 'Someskin2'  
};
```

AutoComplete::Customer###Default

Defines the config options for the autocomplete feature.

Standardwert:

```
$Self->{'AutoComplete::Customer'}->{'Default'} = {  
  'AutoCompleteActive' => '1',  
  'ButtonText' => 'Search',  
  'MaxResultsDisplayed' => '20',  
  'MinQueryLength' => '2',  
  'QueryDelay' => '100'  
};
```

ModernizeCustomerFormFields

Use new type of select and autocomplete fields in customer interface, where applicable (InputFields).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'ModernizeCustomerFormFields'} = '1';
```

Framework → Frontend::Customer::Auth

Customer::AuthModule

Defines the module to authenticate customers.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

Customer::AuthModule::DB::CryptType

If "DB" was selected for Customer::AuthModule, the encryption type of passwords must be specified.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Customer::AuthModule::DB::CryptType'} = 'sha2';
```

Customer::AuthModule::DB::Table

If "DB" was selected for Customer::AuthModule, the name of the table where your customer data should be stored must be specified.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Customer::AuthModule::DB::Table'} = 'customer_user';
```

Customer::AuthModule::DB::CustomerKey

If "DB" was selected for Customer::AuthModule, the name of the column for the CustomerKey in the customer table must be specified.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Customer::AuthModule::DB::CustomerKey'} = 'login';
```

Customer::AuthModule::DB::CustomerPassword

If "DB" was selected for Customer::AuthModule, the column name for the CustomerPassword in the customer table must be specified.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Customer::AuthModule::DB::CustomerPassword'} = 'pw';
```

Customer::AuthModule::DB::DSN

If "DB" was selected for Customer::AuthModule, the DSN for the connection to the customer table must be specified.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::DB::DSN'} =  
'DBI:mysql:database=customerdb;host=customerdbhost';
```

Customer::AuthModule::DB::User

If "DB" was selected for Customer::AuthModule, a username to connect to the customer table can be specified.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::DB::User'} = 'some_user';
```

Customer::AuthModule::DB::Password

If "DB" was selected for Customer::AuthModule, a password to connect to the customer table can be specified.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::DB::Password'} = 'some_password';
```

Customer::AuthModule::DB::Type

If "DB" was selected for Customer::AuthModule, a database driver (normally autodetection is used) can be specified.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::DB::Type'} = 'mysql';
```

Customer::AuthModule::HTTPBasicAuth::Replace

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).

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::HTTPBasicAuth::Replace'} = 'example_domain\\';
```

Customer::AuthModule::HTTPBasicAuth::ReplaceRegExp

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). Reg-Exp-Note, \$1 will be the new Login.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::HTTPBasicAuth::ReplaceRegExp'} = '^(.+?)@.+?$';
```

Customer::AuthModule::LDAP::Host

If "LDAP" was selected for Customer::AuthModule, the LDAP host can be specified.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::LDAP::Host'} = 'ldap.example.com';
```

Customer::AuthModule::LDAP::BaseDN

If "LDAP" was selected for Customer::AuthModule, the BaseDN must be specified.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::LDAP::BaseDN'} = 'dc=example,dc=com';
```

Customer::AuthModule::LDAP::UID

If "LDAP" was selected for Customer::AuthModule, the user identifier must be specified.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::LDAP::UID'} = 'uid';
```

Customer::AuthModule::LDAP::GroupDN

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::LDAP::GroupDN'} =  
'cn=otrsallow,ou=posixGroups,dc=example,dc=com';
```

Customer::AuthModule::LDAP::AccessAttr

If "LDAP" was selected for Customer::AuthModule, you can specify access attributes here.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::LDAP::AccessAttr'} = 'memberUid';
```

Customer::AuthModule::LDAP::UserAttr

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::LDAP::UserAttr'} = 'UID';
```

Customer::AuthModule::LDAP::SearchUserDN

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::LDAP::SearchUserDN'} =  
'cn=binduser,ou=users,dc=example,dc=com';
```

Customer::AuthModule::LDAP::SearchUserPw

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::LDAP::SearchUserPw'} = 'some_password';
```

Customer::AuthModule::LDAP::AlwaysFilter

If "LDAP" was selected, you can add a filter to each LDAP query, e.g. (mail=*), (objectclass=user) or (!objectclass=computer).

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::LDAP::AlwaysFilter'} = '(!objectclass=computer)';
```

Customer::AuthModule::LDAP::UserSuffix

If "LDAP" was selected for Customer::AuthModule and if you want to add a suffix to every customer login name, specify it here, e. g. you just want to write the username user but in your LDAP directory exists user@domain.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::LDAP::UserSuffix'} = '@domain.com';
```

Customer::AuthModule::LDAP::Params

If "LDAP" was selected for Customer::AuthModule and special parameters are needed for the Net::LDAP perl module, you can specify them here. See "perldoc Net::LDAP" for more information about the parameters.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::LDAP::Params'} = {  
  'async' => '0',  
  'port' => '389',  
  'timeout' => '120',  
  'version' => '3'
```

```
};
```

Customer::AuthModule::LDAP::Die

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.

Standardwert:

```
$Self->{'Customer::AuthModule::LDAP::Die'} = '1';
```

Customer::AuthModule::Radius::Host

If "Radius" was selected for Customer::AuthModule, the radius host must be specified.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::Radius::Host'} = 'radiushost';
```

Customer::AuthModule::Radius::Password

If "Radius" was selected for Customer::AuthModule, the password to authenticate to the radius host must be specified.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthModule::Radius::Password'} = 'radiussecret';
```

Customer::AuthModule::Radius::Die

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.

Standardwert:

```
$Self->{'Customer::AuthModule::Radius::Die'} = '1';
```

Framework → Frontend::Customer::Auth::TwoFactor

Customer::AuthTwoFactorModule

Defines the two-factor module to authenticate customers.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Customer::AuthTwoFactorModule'} =  
'Kernel::System::CustomerAuth::TwoFactor::GoogleAuthenticator';
```

Customer::AuthTwoFactorModule::SecretPreferencesKey

Defines the customer preferences key where the shared secret key is stored.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Customer::AuthTwoFactorModule::SecretPreferencesKey'} =  
'UserGoogleAuthenticatorSecretKey';
```

Customer::AuthTwoFactorModule::AllowEmptySecret

Defines if customers should be allowed to login if they have no shared secret stored in their preferences and therefore are not using two-factor authentication.

Standardwert:

```
$Self->{'Customer::AuthTwoFactorModule::AllowEmptySecret'} = '1';
```

Customer::AuthTwoFactorModule::AllowPreviousToken

Defines if the previously valid token should be accepted for authentication. This is slightly less secure but gives users 30 seconds more time to enter their one-time password.

Standardwert:

```
$Self->{'Customer::AuthTwoFactorModule::AllowPreviousToken'} = '1';
```

Framework → Frontend::Customer::ModuleMetaHead

CustomerFrontend::HeaderMetaModule###1-Refresh

Defines the module to generate code for periodic page reloads.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerFrontend::HeaderMetaModule'}->{'1-Refresh'} = {
  'Module' => 'Kernel::Output::HTML::HeaderMeta::Refresh'
};
```

Framework → Frontend::Customer::ModuleNotify

CustomerFrontend::NotifyModule###1-OTRSBusiness

Defines the module to display a notification in different interfaces on different occasions for OTRS Business Solution™.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerFrontend::NotifyModule'}->{'1-OTRSBusiness'} = {
  'Module' => 'Kernel::Output::HTML::Notification::CustomerOTRSBusiness'
};
```

CustomerFrontend::NotifyModule###1-ShowAgentOnline

Defines the module that shows the currently logged in agents in the customer interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'CustomerFrontend::NotifyModule'}->{'1-ShowAgentOnline'} = {
  'IdleMinutes' => '60',
  'Module' => 'Kernel::Output::HTML::Notification::AgentOnline',
  'ShowEmail' => '1'
};
```

CustomerFrontend::NotifyModule###1-ShowCustomerOnline

Defines the module that shows the currently logged in customers in the customer interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'CustomerFrontend::NotifyModule'}->{'1-ShowCustomerOnline'} = {
  'Module' => 'Kernel::Output::HTML::Notification::CustomerOnline',
  'ShowEmail' => '1'
};
```

CustomerFrontend::NotifyModule###6-CustomerSystemMaintenance-Check

Defines the module to display a notification in the customer interface, if the customer is logged in while having system maintenance active.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerFrontend::NotifyModule'}->{'6-CustomerSystemMaintenance-Check'} = {
  'Module' => 'Kernel::Output::HTML::Notification::CustomerSystemMaintenanceCheck'
};
```

CustomerFrontend::NotifyModule###7-CustomerUserTimeZone-Check

Defines the module to display a notification in the customer interface, if the customer user has not yet selected a time zone.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerFrontend::NotifyModule'}->{'7-CustomerUserTimeZone-Check'} = {
  'Module' => 'Kernel::Output::HTML::Notification::CustomerUserTimeZoneCheck'
};
```

Framework → Frontend::Customer::ModuleRegistration

CustomerFrontend::Module###Logout

Frontend module registration for the customer interface.

Standardwert:

```
$Self->{'CustomerFrontend::Module'}->{'Logout'} = {
  'Description' => 'Logout of customer panel.',
  'NavBarName' => '',
  'Title' => ''
};
```

CustomerFrontend::Module###CustomerPreferences

Frontend module registration for the customer interface.

Standardwert:

```
$Self->{'CustomerFrontend::Module'}->{'CustomerPreferences'} = {
  'Description' => 'Customer preferences.',
  'NavBarName' => '',
  'Title' => 'Preferences'
};
```

CustomerFrontend::Module###CustomerAccept

Frontend module registration for the customer interface.

Standardwert:

```
$Self->{'CustomerFrontend::Module'}->{'CustomerAccept'} = {
  'Description' => 'To accept login information, such as an EULA or license.',
  'NavBarName' => '',
  'Title' => 'Info'
};
```

CustomerFrontend::Module###PictureUpload

Frontend module registration for the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerFrontend::Module'}->{'PictureUpload'} = {
  'Description' => 'Picture upload module.',
  'NavBarName' => 'Ticket',
  'Title' => 'Picture-Upload'
};
```

Framework → Frontend::Customer::Preferences

CustomerPreferences

Defines the parameters for the customer preferences table.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerPreferences'} = {
  'Module' => 'Kernel::System::CustomerUser::Preferences::DB',
  'Params' => {
    'Table' => 'customer_preferences',
    'TableKey' => 'preferences_key',
    'TableUserID' => 'user_id',
    'TableValue' => 'preferences_value'
  }
};
```

CustomerPreferencesView

Sets the order of the different items in the customer preferences view.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerPreferencesView'} = [
  'User Profile',
  'Other Settings'
];
```

CustomerPreferencesGroups###Password

Defines all the parameters for this item in the customer preferences. 'PasswordRegExp' allows to match passwords against a regular expression. Define the minimum number of characters using 'PasswordMinSize'. Define if at least 2 lowercase and 2 uppercase letter characters are needed by setting the appropriate option to '1'. 'PasswordMin2Characters' defines if the password needs to contain at least 2 letter characters (set to 0 or 1). 'PasswordNeedDigit' controls the need of at least 1 digit (set to 0 or 1 to control).

Standardwert:

```
$Self->{'CustomerPreferencesGroups'}->{'Password'} = {
  'Active' => '1',
  'Area' => 'Customer',
  'Column' => 'Other Settings',
  'Label' => 'Change password',
  'Module' => 'Kernel::Output::HTML::Preferences::Password',
  'PasswordMin2Characters' => '0',
  'PasswordMin2Lower2UpperCharacters' => '0',
  'PasswordMinSize' => '0',
  'PasswordNeedDigit' => '0',
  'PasswordRegExp' => '',
  'Prio' => '1000'
};
```

CustomerPreferencesGroups###GoogleAuthenticatorSecretKey

Defines the config parameters of this item, to be shown in the preferences view.

Standardwert:

```
$Self->{'CustomerPreferencesGroups'}->{'GoogleAuthenticatorSecretKey'} = {
  'Active' => '0',
  'Block' => 'Input',
  'Column' => 'Other Settings',
  'Key' => 'Shared Secret',
  'Label' => 'Google Authenticator',
};
```



```
'Module' => 'Kernel::Output::HTML::Preferences::Generic',
'PrefKey' => 'UserGoogleAuthenticatorSecretKey',
'Prio' => '1100'
};
```

CustomerPreferencesGroups###Language

Defines all the parameters for this item in the customer preferences.

Standardwert:

```
$Self->{'CustomerPreferencesGroups'}->{'Language'} = {
'Active' => '1',
'Column' => 'User Profile',
'Key' => 'Language',
'Label' => 'Interface language',
'Module' => 'Kernel::Output::HTML::Preferences::Language',
'PrefKey' => 'UserLanguage',
'Prio' => '2000'
};
```

CustomerPreferencesGroups###TimeZone

Defines all the parameters for this item in the customer preferences.

Standardwert:

```
$Self->{'CustomerPreferencesGroups'}->{'TimeZone'} = {
'Active' => '1',
'Column' => 'User Profile',
'Key' => 'Time Zone',
'Label' => 'Time Zone',
'Module' => 'Kernel::Output::HTML::Preferences::TimeZone',
'PrefKey' => 'UserTimeZone',
'Prio' => '2500'
};
```

CustomerPreferencesGroups###Theme

Defines all the parameters for this item in the customer preferences.

Standardwert:

```
$Self->{'CustomerPreferencesGroups'}->{'Theme'} = {
'Active' => '0',
'Column' => 'User Profile',
'Key' => 'Select your frontend Theme.',
'Label' => 'Theme',
'Module' => 'Kernel::Output::HTML::Preferences::Theme',
'PrefKey' => 'UserTheme',
'Prio' => '1000'
};
```

CustomerPreferencesGroups###PGP

Defines all the parameters for this item in the customer preferences.

Standardwert:

```
$Self->{'CustomerPreferencesGroups'}->{'PGP'} = {
'Active' => '1',
'Column' => 'Other Settings',
'Key' => 'PGP Key Upload',
'Label' => 'PGP Key',
'Module' => 'Kernel::Output::HTML::Preferences::PGP',
'PrefKey' => 'UserPGPKey',
'Prio' => '10000'
};
```

CustomerPreferencesGroups###SMIME

Defines all the parameters for this item in the customer preferences.

Standardwert:

```
$Self->{'CustomerPreferencesGroups'}->{'SMIME'} = {
  'Active' => '1',
  'Column' => 'Other Settings',
  'Key' => 'S/MIME Certificate Upload',
  'Label' => 'S/MIME Certificate',
  'Module' => 'Kernel::Output::HTML::Preferences::SMIME',
  'PrefKey' => 'UserSMIMEKey',
  'Prio' => '11000'
};
```

Framework → Frontend::Public

PublicFrontend::CommonParam###Action

Defines the default value for the action parameter for the public frontend. The action parameter is used in the scripts of the system.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PublicFrontend::CommonParam'}->{'Action'} = 'PublicDefault';
```

Framework → Frontend::Public::ModuleRegistration

PublicFrontend::Module###PublicDefault

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'PublicFrontend::Module'}->{'PublicDefault'} = {
  'Description' => 'PublicDefault',
  'NavBarName' => '',
  'Title' => 'PublicDefault'
};
```

PublicFrontend::Module###PublicRepository

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'PublicFrontend::Module'}->{'PublicRepository'} = {
  'Description' => 'PublicRepository',
  'NavBarName' => '',
  'Title' => 'PublicRepository'
};
```

PublicFrontend::Module###PublicSupportDataCollector

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'PublicFrontend::Module'}->{'PublicSupportDataCollector'} = {
  'Description' => 'PublicSupportDataCollector',
  'NavBarName' => '',
  'Title' => 'PublicSupportDataCollector'
};
```

Framework → SystemMaintenance

SystemMaintenance::TimeNotifyUpcomingMaintenance

Sets the minutes a notification is shown for notice about upcoming system maintenance period.

Standardwert:

```
$Self->{'SystemMaintenance::TimeNotifyUpcomingMaintenance'} = '30';
```

SystemMaintenance::IsActiveDefaultNotification

Sets the default message for the notification is shown on a running system maintenance period.

Standardwert:

```
$Self->{'SystemMaintenance::IsActiveDefaultNotification'} = 'We are performing scheduled maintenance.';
```

SystemMaintenance::IsActiveDefaultLoginMessage

Sets the default message for the login screen on Agent and Customer interface, it's shown when a running system maintenance period is active.

Standardwert:

```
$Self->{'SystemMaintenance::IsActiveDefaultLoginMessage'} = 'We are performing scheduled maintenance. We should be back online shortly.';
```

SystemMaintenance::IsActiveDefaultLoginErrorMessage

Sets the default error message for the login screen on Agent and Customer interface, it's shown when a running system maintenance period is active.

Standardwert:

```
$Self->{'SystemMaintenance::IsActiveDefaultLoginErrorMessage'} = 'We are performing scheduled maintenance. Login is temporarily not available.';
```

5. GenericInterface

GenericInterface → Core::CustomerCompany

CustomerCompany::EventModulePost###9900-GenericInterface

Performs the configured action for each event (as an Invoker) for each configured Webservice.

Standardwert:

```
$Self->{'CustomerCompany::EventModulePost'}->{'9900-GenericInterface'} = {
  'Event' => '',
  'Module' => 'Kernel::GenericInterface::Event::Handler',
  'Transaction' => '1'
};
```

GenericInterface → Core::CustomerUser

CustomerUser::EventModulePost###9900-GenericInterface

Performs the configured action for each event (as an Invoker) for each configured Webservice.

Standardwert:

```
$Self->{'CustomerUser::EventModulePost'}->{'9900-GenericInterface'} = {
  'Event' => '',
  'Module' => 'Kernel::GenericInterface::Event::Handler',
  'Transaction' => '1'
};
```

GenericInterface → Core::DynamicField

DynamicField::EventModulePost###9900-GenericInterface

Performs the configured action for each event (as an Invoker) for each configured Webservice.

Standardwert:

```
$Self->{'DynamicField::EventModulePost'}->{'9900-GenericInterface'} = {
  'Event' => '',
  'Module' => 'Kernel::GenericInterface::Event::Handler',
  'Transaction' => '1'
};
```

GenericInterface → Core::Package

Package::EventModulePost###9900-GenericInterface

Performs the configured action for each event (as an Invoker) for each configured Webservice.

Standardwert:

```
$Self->{'Package::EventModulePost'}->{'9900-GenericInterface'} = {
  'Event' => '',
  'Module' => 'Kernel::GenericInterface::Event::Handler',
  'Transaction' => '1'
};
```

GenericInterface → Core::Queue

Queue::EventModulePost###9900-GenericInterface

Performs the configured action for each event (as an Invoker) for each configured Webservice.

Standardwert:

```
$Self->{'Queue::EventModulePost'}->{'9900-GenericInterface'} = {
  'Event' => '',
  'Module' => 'Kernel::GenericInterface::Event::Handler',
  'Transaction' => '1'
};
```

GenericInterface → Core::Ticket

Ticket::EventModulePost###9900-GenericInterface

Performs the configured action for each event (as an Invoker) for each configured Webservice.

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'9900-GenericInterface'} = {
  'Event' => '',
  'Module' => 'Kernel::GenericInterface::Event::Handler',
  'Transaction' => '1'
};
```

GenericInterface → Frontend::Admin::ModuleRegistration

Frontend::Module###AdminGenericInterfaceDebugger

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminGenericInterfaceDebugger'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.GenericInterface.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.GenericInterfaceDebugger.js'
    ]
  }
};
```

```

    ],
  },
  'Title' => 'GenericInterface Debugger GUI'
};

```

Frontend::Module###AdminGenericInterfaceWebservice

Frontend module registration for the agent interface.

Standardwert:

```

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

```

Frontend::Module###AdminGenericInterfaceTransportHTTPSOAP

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AdminGenericInterfaceTransportHTTPSOAP'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.GenericInterface.css',
      'Core.Agent.SortedTree.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.GenericInterfaceTransportHTTPSOAP.js',
      'Core.Agent.SortedTree.js'
    ]
  },
  'Title' => 'GenericInterface TransportHTTPSOAP GUI'
};

```

Frontend::Module###AdminGenericInterfaceTransportHTTPREST

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AdminGenericInterfaceTransportHTTPREST'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.GenericInterface.css'
    ]
  }
};

```

```

    ],
    'JavaScript' => [
      'Core.Agent.Admin.GenericInterfaceTransportHTTPREST.js'
    ]
  },
  'Title' => 'GenericInterface TransportHTTPREST GUI'
};

```

Frontend::Module###AdminGenericInterfaceWebserviceHistory

Frontend module registration for the agent interface.

Standardwert:

```

$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'
};

```

Frontend::Module###AdminGenericInterfaceOperationDefault

Frontend module registration for the agent interface.

Standardwert:

```

$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'
};

```

Frontend::Module###AdminGenericInterfaceInvokerDefault

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AdminGenericInterfaceInvokerDefault'} = {
  'Description' => 'Admin',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.GenericInterface.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.GenericInterfaceInvoker.js'
    ]
  },
  'Title' => 'GenericInterface Invoker GUI'
};

```

Frontend::Module###AdminGenericInterfaceMappingSimple

Frontend module registration for the agent interface.

Standardwert:

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

Frontend::Module###AdminGenericInterfaceMappingXSLT

Frontend module registration for the agent interface.

Standardwert:

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

GenericInterface

GenericInterface::Invoker::ModuleRegistration

GenericInterface::Invoker::Module###Test::Test

GenericInterface module registration for the invoker layer.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'GenericInterface::Invoker::Module'}->{'Test::Test'} = {
  'ConfigDialog' => 'AdminGenericInterfaceInvokerDefault',
  'Controller' => 'Test',
  'Name' => 'Test'
};
```

GenericInterface::Invoker::Module###Test::TestSimple

GenericInterface module registration for the invoker layer.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'GenericInterface::Invoker::Module'}->{'Test::TestSimple'} = {
  'ConfigDialog' => 'AdminGenericInterfaceInvokerDefault',
  'Controller' => 'Test',
};
```

```
'Name' => 'TestSimple'
};
```

GenericInterface



GenericInterface::Mapping::ModuleRegistration

GenericInterface::Mapping::Module###Test

GenericInterface module registration for the mapping layer.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'GenericInterface::Mapping::Module'}->{'Test'} = {
  'ConfigDialog' => ''
};
```

GenericInterface::Mapping::Module###Simple

GenericInterface module registration for the mapping layer.

Standardwert:

```
$Self->{'GenericInterface::Mapping::Module'}->{'Simple'} = {
  'ConfigDialog' => 'AdminGenericInterfaceMappingSimple'
};
```

GenericInterface::Mapping::Module###XSLT

GenericInterface module registration for the mapping layer.

Standardwert:

```
$Self->{'GenericInterface::Mapping::Module'}->{'XSLT'} = {
  'ConfigDialog' => 'AdminGenericInterfaceMappingXSLT'
};
```

GenericInterface



GenericInterface::Operation::ModuleRegistration

GenericInterface::Operation::Module###Test::Test

GenericInterface module registration for the operation layer.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'GenericInterface::Operation::Module'}->{'Test::Test'} = {
  'ConfigDialog' => 'AdminGenericInterfaceOperationDefault',
  'Controller' => 'Test',
  'Name' => 'Test'
};
```

GenericInterface::Operation::Module###Session::SessionCreate

GenericInterface module registration for the operation layer.

Standardwert:

```
$Self->{'GenericInterface::Operation::Module'}->{'Session::SessionCreate'} = {
  'ConfigDialog' => 'AdminGenericInterfaceOperationDefault',
  'Controller' => 'Session',
  'Name' => 'SessionCreate'
};
```

GenericInterface::Operation::Module###Ticket::TicketCreate

GenericInterface module registration for the operation layer.

Standardwert:


```
$Self->{'GenericInterface::Operation::Module'}->{'Ticket::TicketCreate'} = {
  'ConfigDialog' => 'AdminGenericInterfaceOperationDefault',
  'Controller' => 'Ticket',
  'Name' => 'TicketCreate'
};
```

GenericInterface::Operation::Module###Ticket::TicketUpdate

GenericInterface module registration for the operation layer.

Standardwert:

```
$Self->{'GenericInterface::Operation::Module'}->{'Ticket::TicketUpdate'} = {
  'ConfigDialog' => 'AdminGenericInterfaceOperationDefault',
  'Controller' => 'Ticket',
  'Name' => 'TicketUpdate'
};
```

GenericInterface::Operation::Module###Ticket::TicketGet

GenericInterface module registration for the operation layer.

Standardwert:

```
$Self->{'GenericInterface::Operation::Module'}->{'Ticket::TicketGet'} = {
  'ConfigDialog' => 'AdminGenericInterfaceOperationDefault',
  'Controller' => 'Ticket',
  'Name' => 'TicketGet'
};
```

GenericInterface::Operation::Module###Ticket::TicketSearch

GenericInterface module registration for the operation layer.

Standardwert:

```
$Self->{'GenericInterface::Operation::Module'}->{'Ticket::TicketSearch'} = {
  'ConfigDialog' => 'AdminGenericInterfaceOperationDefault',
  'Controller' => 'Ticket',
  'Name' => 'TicketGet'
};
```

GenericInterface

GenericInterface::Operation::ResponseLoggingMaxSize →

GenericInterface::Operation::ResponseLoggingMaxSize

Defines the maximum size in KiloByte of GenericInterface responses that get logged to the gi_debugger_entry_content table.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'GenericInterface::Operation::ResponseLoggingMaxSize'} = '200';
```

GenericInterface → GenericInterface::Operation::TicketCreate

GenericInterface::Operation::TicketCreate###ArticleType

Defines the default type of the article for this operation.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'GenericInterface::Operation::TicketCreate'}->{'ArticleType'} = 'webrequest';
```

GenericInterface::Operation::TicketCreate###HistoryType

Defines the history type for this operation, which gets used for ticket history in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'GenericInterface::Operation::TicketCreate'}->{'HistoryType'} = 'NewTicket';
```

GenericInterface::Operation::TicketCreate###HistoryComment

Defines the history comment for this operation, which gets used for ticket history in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'GenericInterface::Operation::TicketCreate'}->{'HistoryComment'} = '%  
%GenericInterface Create';
```

GenericInterface::Operation::TicketCreate###AutoResponseType

Defines the default auto response type of the article for this operation.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'GenericInterface::Operation::TicketCreate'}->{'AutoResponseType'} = 'auto  
reply';
```

GenericInterface → GenericInterface::Operation::TicketSearch

GenericInterface::Operation::TicketSearch###SearchLimit

Maximum number of tickets to be displayed in the result of this operation.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'GenericInterface::Operation::TicketSearch'}->{'SearchLimit'} = '500';
```

GenericInterface::Operation::TicketSearch###SortBy::Default

Defines the default ticket attribute for ticket sorting of the ticket search result of this operation.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'GenericInterface::Operation::TicketSearch'}->{'SortBy::Default'} = 'Age';
```

GenericInterface::Operation::TicketSearch###Order::Default

Defines the default ticket order in the ticket search result of the this operation. Up: oldest on top. Down: latest on top.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'GenericInterface::Operation::TicketSearch'}->{'Order::Default'} = 'Down';
```

GenericInterface → GenericInterface::Operation::TicketUpdate

GenericInterface::Operation::TicketUpdate###ArticleType

Defines the default type of the article for this operation.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'GenericInterface::Operation::TicketUpdate'}->{'ArticleType'} = 'webrequest';
```

GenericInterface::Operation::TicketUpdate###HistoryType

Defines the history type for this operation, which gets used for ticket history in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'GenericInterface::Operation::TicketUpdate'}->{'HistoryType'} = 'AddNote';
```

GenericInterface::Operation::TicketUpdate###HistoryComment

Defines the history comment for this operation, which gets used for ticket history in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'GenericInterface::Operation::TicketUpdate'}->{'HistoryComment'} = '%GenericInterface Note';
```

GenericInterface::Operation::TicketUpdate###AutoResponseType

Defines the default auto response type of the article for this operation.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'GenericInterface::Operation::TicketUpdate'}->{'AutoResponseType'} = 'auto follow up';
```

GenericInterface

GenericInterface::Transport::ModuleRegistration

→

GenericInterface::Transport::Module###HTTP::SOAP

GenericInterface module registration for the transport layer.

Standardwert:

```
$Self->{'GenericInterface::Transport::Module'}->{'HTTP::SOAP'} = {
  'ConfigDialog' => 'AdminGenericInterfaceTransportHTTPSOAP',
  'Name' => 'SOAP',
  'Protocol' => 'HTTP'
};
```

GenericInterface::Transport::Module###HTTP::REST

GenericInterface module registration for the transport layer.

Standardwert:

```
$Self->{'GenericInterface::Transport::Module'}->{'HTTP::REST'} = {
  'ConfigDialog' => 'AdminGenericInterfaceTransportHTTPREST',
  'Name' => 'REST',
  'Protocol' => 'HTTP'
};
```

GenericInterface::Transport::Module###HTTP::Test

GenericInterface module registration for the transport layer.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'GenericInterface::Transport::Module'}->{'HTTP::Test'} = {
  'ConfigDialog' => 'AdminGenericInterfaceTransportHTTPTest',
  'Name' => 'Test',
  'Protocol' => 'HTTP'
};
```

GenericInterface → GenericInterface::Webservice

GenericInterface::WebserviceConfig::CacheTTL

Cache time in seconds for the web service config backend.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'GenericInterface::WebserviceConfig::CacheTTL'} = '86400';
```

GenericInterface::Operation::Common::CachedAuth::AgentCacheTTL

Cache time in seconds for agent authentication in the GenericInterface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'GenericInterface::Operation::Common::CachedAuth::AgentCacheTTL'} = '300';
```

GenericInterface::Operation::Common::CachedAuth::CustomerCacheTTL

Cache time in seconds for customer authentication in the GenericInterface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'GenericInterface::Operation::Common::CachedAuth::CustomerCacheTTL'} = '300';
```

6. ProcessManagement

ProcessManagement → Core

Process::DynamicFieldProcessManagementProcessID

This option defines the dynamic field in which a Process Management process entity id is stored.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Process::DynamicFieldProcessManagementProcessID'} =
  'ProcessManagementProcessID';
```

Process::DynamicFieldProcessManagementActivityID

This option defines the dynamic field in which a Process Management activity entity id is stored.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Process::DynamicFieldProcessManagementActivityID'} =
  'ProcessManagementActivityID';
```

Process::DefaultQueue

This option defines the process tickets default queue.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Process::DefaultQueue'} = 'Raw';
```

Process::DefaultState

This option defines the process tickets default state.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Process::DefaultState'} = 'new';
```

Process::DefaultLock

This option defines the process tickets default lock.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Process::DefaultLock'} = 'unlock';
```

Process::DefaultPriority

This option defines the process tickets default priority.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Process::DefaultPriority'} = '3 normal';
```

Process::Entity::Prefix

Default ProcessManagement entity prefixes for entity IDs that are automatically generated.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Process::Entity::Prefix'} = {  
  'Activity' => 'A',  
  'ActivityDialog' => 'AD',  
  'Process' => 'P',  
  'Transition' => 'T',  
  'TransitionAction' => 'TA'  
};
```

Process::CacheTTL

Cache time in seconds for the DB process backend.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Process::CacheTTL'} = '3600';
```

Process::NavBarOutput::CacheTTL

Cache time in seconds for the ticket process navigation bar output module.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Process::NavBarOutput::CacheTTL'} = '900';
```

ProcessManagement → Core::Ticket

Ticket::EventModulePost###9800-TicketProcessTransitions

Event module registration. For more performance you can define a trigger event (e.g. Event => TicketCreate).

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'9800-TicketProcessTransitions'} = {
  'Event' => '',
  'Module' => 'Kernel::System::Ticket::Event::TicketProcessTransitions',
  'Transaction' => '1'
};
```

ProcessManagement → Core::Transition

ProcessManagement::Transition::Debug::Enabled

If enabled debugging information for transitions is logged.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'ProcessManagement::Transition::Debug::Enabled'} = '0';
```

ProcessManagement::Transition::Debug::LogPriority

Defines the priority in which the information is logged and presented.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'ProcessManagement::Transition::Debug::LogPriority'} = 'debug';
```

ProcessManagement::Transition::Debug::Filter###00-Default

Filter for debugging Transitions. Note: More filters can be added in the format <OTRS_TICKET_Attribute> e.g. <OTRS_TICKET_Priority>.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'ProcessManagement::Transition::Debug::Filter'}->{'00-Default'} = {
  '<OTRS_TICKET_TicketNumber>' => '',
  'TransitionEntityID' => ''
};
```

ProcessManagement → DynamicFields::Driver::Registration

DynamicFields::Driver###ProcessID

DynamicField backend registration.

Standardwert:

```
$Self->{'DynamicFields::Driver'}->{'ProcessID'} = {
  'ConfigDialog' => 'AdminDynamicFieldText',
  'DisabledAdd' => '1',
  'DisplayName' => 'ProcessID',
  'Module' => 'Kernel::System::DynamicField::Driver::ProcessManagement::ProcessID'
};
```

DynamicFields::Driver###ActivityID

DynamicField backend registration.

Standardwert:

```
$Self->{'DynamicFields::Driver'}->{'ActivityID'} = {
```

```
'ConfigDialog' => 'AdminDynamicFieldText',
'DisabledAdd' => '1',
'DisplayName' => 'ActivityID',
'Module' => 'Kernel::System::DynamicField::Driver::ProcessManagement::ActivityID'
};
```

ProcessManagement → Frontend::Admin::ModuleRegistration

Frontend::Module###AdminProcessManagement

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminProcessManagement'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ProcessManagement.css',
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'thirdparty/jsplumb-1.6.4/jsplumb.js',
      'thirdparty/farahey-0.5/farahey.js',
      'thirdparty/jsplumb-labelspacer/label-spacer.js',
      'Core.Agent.Admin.ProcessManagement.js',
      'Core.Agent.Admin.ProcessManagement.Canvas.js',
      'Core.UI.AllocationList.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Configure Processes.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Process Management',
    'Prio' => '750'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Process Management'
};
```

Frontend::Module###AdminProcessManagementActivity

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminProcessManagementActivity'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ProcessManagement.css',
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ProcessManagement.js',
      'Core.UI.AllocationList.js'
    ]
  },
  'Title' => 'Process Management Activity GUI'
};
```

Frontend::Module###AdminProcessManagementActivityDialog

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminProcessManagementActivityDialog'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ProcessManagement.css',
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ProcessManagement.js',
      'Core.UI.AllocationList.js'
    ]
  },
  'Title' => 'Process Management Activity Dialog GUI'
};
```

Frontend::Module###AdminProcessManagementTransition

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminProcessManagementTransition'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ProcessManagement.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ProcessManagement.js'
    ]
  },
  'Title' => 'Process Management Transition GUI'
};
```

Frontend::Module###AdminProcessManagementTransitionAction

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminProcessManagementTransitionAction'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ProcessManagement.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ProcessManagement.js'
    ]
  },
  'Title' => 'Process Management Transition Action GUI'
};
```

Frontend::Module###AdminProcessManagementPath

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminProcessManagementPath'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
```



```

    'CSS' => [
      'Core.Agent.Admin.ProcessManagement.css',
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ProcessManagement.js',
      'Core.UI.AllocationList.js'
    ]
  },
  'Title' => 'Process Management Path GUI'
};

```

ProcessManagement → Frontend::Agent::Dashboard

DashboardBackend###0140-RunningTicketProcess

Parameters for the dashboard backend of the running process tickets overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.

Standardwert:

```

$self->{'DashboardBackend'}->{'0140-RunningTicketProcess'} = {
  'Attributes' => 'StateType=new;StateType=open;StateType=pending
reminder;StateType=pending auto',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '0',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'DynamicField_ProcessManagementActivityID' => '2',
    'DynamicField_ProcessManagementProcessID' => '2',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All tickets with a reminder set where the reminder date has been
reached',
  'Group' => '',
  'IsProcessWidget' => '1',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::Dashboard::TicketGeneric',
  'Permission' => 'rw',
  'Time' => 'UntilTime',
  'Title' => 'Running Process Tickets'
};

```

ProcessManagement → Frontend::Agent::ModuleRegistration

Frontend::Module###AgentTicketProcess

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentTicketProcess'} = {
  'Description' => 'Create new process ticket.',
  'Loader' => {
    'CSS' => [
      'Core.Agent.TicketProcess.css'
    ],
    'JavaScript' => [
      'Core.Agent.CustomerSearch.js',
      'Core.Agent.CustomerSearchAutoComplete.js',
      'Core.Agent.TicketAction.js',
      'Core.Agent.TicketProcess.js',
      'Core.TicketProcess.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'p',
      'Block' => '',
      'Description' => 'Create New process ticket.',
      'Link' => 'Action=AgentTicketProcess',
      'LinkOption' => '',
      'Name' => 'New process ticket',
      'NavBar' => 'Ticket',
      'Prio' => '220',
      'Type' => ''
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'New process ticket'
};
```

ProcessManagement → Frontend::Agent::NavBarModule

Frontend::NavBarModule###1-TicketProcesses

Frontend module registration (disable ticket processes screen if no process available).

Standardwert:

```
$Self->{'Frontend::NavBarModule'}->{'1-TicketProcesses'} = {
  'Module' => 'Kernel::Output::HTML::NavBar::AgentTicketProcess'
};
```

ProcessManagement → Frontend::Agent::Ticket::MenuModule

Ticket::Frontend::MenuModule###480-Process

Shows a link in the menu to enroll a ticket into a process in the ticket zoom view of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'480-Process'} = {
  'Action' => 'AgentTicketProcess',
  'Cluster' => '',
  'Description' => 'Enroll process for this ticket',
  'Link' => 'Action=AgentTicketProcess;IsProcessEnroll=1;TicketID=[% Data.TicketID |
html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Process',
  'Name' => 'Process',
  'PopupType' => 'TicketAction',
  'Target' => ''
};
```

ProcessManagement → Frontend::Agent::Ticket::ViewProcess

Ticket::Frontend::AgentTicketProcess###StateType

Determines the next possible ticket states, for process tickets in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketProcess'}->{'StateType'} = [
  'new',
  'open',
  'pending auto',
  'pending reminder',
  'closed'
];
```

Ticket::Frontend::CustomerTicketProcess###StateType

Determines the next possible ticket states, for process tickets in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketProcess'}->{'StateType'} = [
  'new',
  'open'
];
```

Ticket::Frontend::AgentTicketProcess::CustomerIDReadOnly

Controls if CustomerID is editable in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketProcess::CustomerIDReadOnly'} = '1';
```

ProcessManagement → Frontend::Agent::Ticket::ViewZoom

Ticket::Frontend::AgentTicketZoom###ProcessDisplay

Display settings to override defaults for Process Tickets.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketZoom'}->{'ProcessDisplay'} = {
  'NavBarName' => 'Processes',
  'WidgetTitle' => 'Process Information'
};
```

Ticket::Frontend::AgentTicketZoom###ProcessWidgetDynamicFieldGroups

Dynamic fields groups for process widget. The key is the name of the group, the value contains the fields to be shown. Example: 'Key => My Group', 'Content: Name_X, NameY'.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketZoom'}->{'ProcessWidgetDynamicFieldGroups'} = {};
```

Ticket::Frontend::AgentTicketZoom###ProcessWidgetDynamicField

Dynamic fields shown in the process widget in ticket zoom screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketZoom'}->{'ProcessWidgetDynamicField'} = {};
```

ProcessManagement → Frontend::Customer::ModuleRegistration

CustomerFrontend::Module###CustomerTicketProcess

Frontend module registration for the customer interface.

Standardwert:

```
$Self->{'CustomerFrontend::Module'}->{'CustomerTicketProcess'} = {
  'Description' => 'Process Ticket.',
  'Loader' => {
    'CSS' => [
      'Core.Customer.TicketProcess.css'
    ],
    'JavaScript' => [
      'Core.TicketProcess.js',
      'Core.Customer.TicketProcess.js',
      'Core.TicketProcess.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'o',
      'Block' => '',
      'Description' => 'Create new process ticket.',
      'Link' => 'Action=CustomerTicketProcess',
      'LinkOption' => '',
      'Name' => 'New process ticket',
      'NavBar' => 'Ticket',
      'Prio' => '220',
      'Type' => 'Submenu'
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'Process ticket'
};
```

ProcessManagement → Frontend::Customer::NavBarModule

CustomerFrontend::NavBarModule###10-CustomerTicketProcesses

Frontend module registration (disable ticket processes screen if no process available) for Customer.

Standardwert:

```
$Self->{'CustomerFrontend::NavBarModule'}->{'10-CustomerTicketProcesses'} = {
  'Module' => 'Kernel::Output::HTML::NavBar::CustomerTicketProcess'
};
```

7. Ticket

Ticket → Core::CustomerCompany

CustomerCompany::EventModulePost###2300-UpdateTickets

Event module that updates tickets after an update of the Customer.

Standardwert:

```
$Self->{'CustomerCompany::EventModulePost'}->{'2300-UpdateTickets'} = {
  'Event' => 'CustomerCompanyUpdate',
  'Module' => 'Kernel::System::CustomerCompany::Event::TicketUpdate',
  'Transaction' => '0'
};
```

CustomerUser::EventModulePost###2300-UpdateTickets

Event module that updates tickets after an update of the Customer User.

Standardwert:

```
$Self->{'CustomerUser::EventModulePost'}->{'2300-UpdateTickets'} = {
  'Event' => 'CustomerUserUpdate',
  'Module' => 'Kernel::System::CustomerUser::Event::TicketUpdate',
  'Transaction' => '0'
};
```

Ticket → Core::FulltextSearch

Ticket::SearchIndexModule

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.Console.pl Maint::Ticket::FulltextIndexRebuild".

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::SearchIndexModule'} =
  'Kernel::System::Ticket::ArticleSearchIndex::RuntimeDB';
```

Ticket::SearchIndex::WarnOnStopWordUsage

Display a warning and prevent search when using stop words within fulltext search.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::SearchIndex::WarnOnStopWordUsage'} = '0';
```

Ticket::SearchIndex::Attribute

Basic fulltext index settings. Execute "bin/otrs.Console.pl Maint::Ticket::FulltextIndexRebuild" in order to generate a new index.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::SearchIndex::Attribute'} = {
  'WordCountMax' => '1000',
  'WordLengthMax' => '30',
  'WordLengthMin' => '3'
};
```

Ticket::SearchIndex::Filters

Fulltext index regex filters to remove parts of the text.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::SearchIndex::Filters'} = [
  '[,\\&\\<\\>\\?\"\\!\\*\\|;\\[\\]\\(\\)\\+\\$\\^=]',
  '^\\[\\.:][\\'\\.:]$',
  '^\\[\\^\\w]+$'
];
```

Ticket::SearchIndex::StopWords###en

English stop words for fulltext index. These words will be removed from the search index.

Standardwert:

```
$Self->{'Ticket::SearchIndex::StopWords'}->{'en'} = [  
  'a',  
  'about',  
  'above',  
  'after',  
  'again',  
  'against',  
  'all',  
  'am',  
  'an',  
  'and',  
  'any',  
  'are',  
  'aren\'t',  
  'as',  
  'at',  
  'be',  
  'because',  
  'been',  
  'before',  
  'being',  
  'below',  
  'between',  
  'both',  
  'but',  
  'by',  
  'can\'t',  
  'cannot',  
  'could',  
  'couldn\'t',  
  'did',  
  'didn\'t',  
  'do',  
  'does',  
  'doesn\'t',  
  'doing',  
  'don\'t',  
  'down',  
  'during',  
  'each',  
  'few',  
  'for',  
  'from',  
  'further',  
  'had',  
  'hadn\'t',  
  'has',  
  'hasn\'t',  
  'have',  
  'haven\'t',  
  'having',  
  'he',  
  'he\'d',  
  'he\'ll',  
  'he\'s',  
  'her',  
  'here',  
  'here\'s',  
  'hers',  
  'herself',  
  'him',  
  'himself',  
  'his',  
  'how',  
  'how\'s',  
  'i',  
  'i\'d',  
  'i\'ll',  
  'i\'m',  
  'i\'ve',
```

'if',
'in',
'into',
'is',
'isn\'t',
'it',
'it\'s',
'its',
'itself',
'let\'s',
'me',
'more',
'most',
'mustn\'t',
'my',
'myself',
'no',
'nor',
'not',
'of',
'off',
'on',
'once',
'only',
'or',
'other',
'ought',
'our',
'ours',
'ourselves',
'out',
'over',
'own',
'same',
'shan\'t',
'she',
'she\'d',
'she\'ll',
'she\'s',
'should',
'shouldn\'t',
'so',
'some',
'such',
'than',
'that',
'that\'s',
'the',
'their',
'theirs',
'them',
'themselves',
'then',
'there',
'there\'s',
'these',
'they',
'they\'d',
'they\'ll',
'they\'re',
'they\'ve',
'this',
'those',
'through',
'to',
'too',
'under',
'until',
'up',
'very',
'was',
'wasn\'t',

```
'we',  
'we\'d',  
'we\'ll',  
'we\'re',  
'we\'ve',  
'were',  
'weren\'t',  
'what',  
'what\'s',  
'when',  
'when\'s',  
'where',  
'where\'s',  
'which',  
'while',  
'who',  
'who\'s',  
'whom',  
'why',  
'why\'s',  
'with',  
'won\'t',  
'would',  
'wouldn\'t',  
'you',  
'you\'d',  
'you\'ll',  
'you\'re',  
'you\'ve',  
'your',  
'yours',  
'yourself',  
'yourselves'  
];
```

Ticket::SearchIndex::StopWords###de

German stop words for fulltext index. These words will be removed from the search index.

Standardwert:

```
$Self->{'Ticket::SearchIndex::StopWords'}->{'de'} = [  
'aber',  
'als',  
'am',  
'an',  
'auch',  
'auf',  
'aus',  
'bei',  
'bin',  
'bis',  
'bist',  
'da',  
'dadurch',  
'daher',  
'darum',  
'das',  
'daß',  
'dass',  
'dein',  
'deine',  
'dem',  
'den',  
'der',  
'des',  
'dessen',  
'deshalb',  
'die',  
'dies',  
'dieser',  
'dieses',
```


'doch',
'dort',
'du',
'durch',
'ein',
'eine',
'einem',
'einen',
'einer',
'eines',
'er',
'es',
'euer',
'eure',
'für',
'hatte',
'hatten',
'hattest',
'hattet',
'hier',
'hinter',
'ich',
'ihr',
'ihre',
'im',
'in',
'ist',
'ja',
'jede',
'jedem',
'jeden',
'jeder',
'jedes',
'jener',
'jenes',
'jetzt',
'kann',
'kannst',
'können',
'könnt',
'machen',
'mein',
'meine',
'mit',
'muß',
'mußt',
'musst',
'müssen',
'müßt',
'nach',
'nachdem',
'nein',
'nicht',
'nun',
'oder',
'seid',
'sein',
'seine',
'sich',
'sie',
'sind',
'soll',
'sollen',
'sollst',
'sollt',
'sonst',
'soweit',
'sowie',
'und',
'unser',
'unsere',
'unter',

```
'vom',  
'von',  
'vor',  
'wann',  
'warum',  
'was',  
'weiter',  
'weitere',  
'wenn',  
'wer',  
'werde',  
'werden',  
'werdet',  
'weshalb',  
'wie',  
'wieder',  
'wieso',  
'wir',  
'wird',  
'wirst',  
'wo',  
'woher',  
'wohin',  
'zu',  
'zum',  
'zur',  
'über'  
];
```

Ticket::SearchIndex::StopWords###nl

Dutch stop words for fulltext index. These words will be removed from the search index.

Standardwert:

```
$Self->{'Ticket::SearchIndex::StopWords'}->{'\n\'} = [  
'de',  
'zijn',  
'een',  
'en',  
'in',  
'je',  
'het',  
'van',  
'op',  
'ze',  
'hebben',  
'het',  
'hij',  
'niet',  
'met',  
'er',  
'dat',  
'die',  
'te',  
'wat',  
'voor',  
'naar',  
'gaan',  
'kunnen',  
'zeggen',  
'dat',  
'maar',  
'aan',  
'veel',  
'zijn',  
'worden',  
'uit',  
'ook',  
'komen',  
'als',  
'om',
```

'moeten',
'we',
'doen',
'bij',
'goed',
'haar',
'dan',
'nog',
'of',
'maken',
'zo',
'wel',
'mijn',
'zien',
'over',
'willen',
'staan',
'door',
'kijken',
'zullen',
'heel',
'nu',
'weten',
'zitten',
'hem',
'schrijven',
'vinden',
'woord',
'hoe',
'geen',
'dit',
'mens',
'al',
'jij',
'ander',
'groot',
'waar',
'maar',
'weer',
'kind',
'me',
'vragen',
'een',
'denken',
'twee',
'horen',
'iets',
'deze',
'krijgen',
'ons',
'zich',
'lezen',
'hun',
'welk',
'zin',
'laten',
'mogen',
'hier',
'jullie',
'toch',
'geven',
'jaar',
'tegen',
'al',
'eens',
'echt',
'houden',
'alleen',
'lopen',
'mee',
'ja',
'roepen',

```
'tijd',
'dag',
'elkaar',
'even',
'lang',
'land',
'liggen',
'waarom',
'zetten',
'vader',
'laat',
'beginnen',
'blijven',
'nee',
'moeder',
'huis',
'nou',
'na',
'af',
'keer',
'dus',
'tot',
'vertellen',
'wie',
'net',
'jou',
'les',
'want',
'man',
'nieuw',
'elk',
'tekst',
'omdat',
'gebruiken',
'u'
];
```

Ticket::SearchIndex::StopWords###es

Spanish stop words for fulltext index. These words will be removed from the search index.

Standardwert:

```
$Self->{'Ticket::SearchIndex::StopWords'}->{'es'} = [
'un',
'una',
'unas',
'unos',
'uno',
'sobre',
'todo',
'también',
'tras',
'otro',
'algún',
'alguno',
'alguna',
'algunos',
'algunas',
'ser',
'es',
'soy',
'eres',
'somos',
'sois',
'estoy',
'esta',
'estamos',
'estais',
'estan',
'como',
'en',
```

'para',
'atras',
'porque',
'por qué',
'estado',
'estaba',
'ante',
'antes',
'siendo',
'ambos',
'pero',
'por',
'poder',
'puede',
'puedo',
'podemos',
'podeis',
'pueden',
'fui',
'fue',
'fuimos',
'fueron',
'hacer',
'hago',
'hace',
'hacemos',
'haceis',
'hacen',
'cada',
'fin',
'incluso',
'primero',
'desde',
'conseguir',
'consigo',
'consigue',
'consigues',
'conseguimos',
'consiguen',
'ir',
'voy',
'va',
'vamos',
'vais',
'van',
'vaya',
'gueno',
'ha',
'tener',
'tengo',
'tiene',
'tenemos',
'teneis',
'tienen',
'el',
'la',
'lo',
'las',
'los',
'su',
'aquí',
'mío',
'tuyo',
'ellos',
'ellas',
'nos',
'nosotros',
'vosotros',
'vosotras',
'sí',
'dentro',
'solo',

'solamente',
'saber',
'sabes',
'sabe',
'sabemos',
'sabeis',
'saben',
'ultimo',
'largo',
'bastante',
'haces',
'muchos',
'aquellos',
'aquellas',
'sus',
'entonces',
'tiempo',
'verdad',
'verdadero',
'verdadera',
'cierto',
'ciertos',
'cierta',
'ciertas',
'intentar',
'intento',
'intenta',
'intentas',
'intentamos',
'intentais',
'intentan',
'dos',
'bajo',
'arriba',
'encima',
'usar',
'uso',
'usas',
'usa',
'usamos',
'usais',
'usan',
'emplear',
'empleo',
'empleas',
'emplean',
'empleamos',
'empleais',
'valor',
'muy',
'era',
'eras',
'eramos',
'eran',
'modo',
'bien',
'cual',
'cuando',
'donde',
'mientras',
'quien',
'con',
'entre',
'sin',
'trabajo',
'trabajar',
'trabajas',
'trabaja',
'trabajamos',
'trabajais',
'trabajan',
'podria',

```
'podrias',  
'podriamos',  
'podrian',  
'podriais',  
'yo',  
'aquel'  
];
```

Ticket::SearchIndex::StopWords###fr

French stop words for fulltext index. These words will be removed from the search index.

Standardwert:

```
$Self->{'Ticket::SearchIndex::StopWords'}->{'fr'} = [  
'alors',  
'au',  
'aucuns',  
'aussi',  
'autre',  
'avant',  
'avec',  
'avoir',  
'bon',  
'car',  
'ce',  
'cela',  
'ces',  
'ceux',  
'chaque',  
'ci',  
'comme',  
'comment',  
'dans',  
'des',  
'du',  
'dedans',  
'dehors',  
'depuis',  
'deux',  
'devrait',  
'doit',  
'donc',  
'dos',  
'droite',  
'début',  
'elle',  
'elles',  
'en',  
'encore',  
'essai',  
'est',  
'et',  
'eu',  
'fait',  
'faites',  
'fois',  
'font',  
'force',  
'haut',  
'hors',  
'ici',  
'il',  
'ils',  
'je',  
'juste',  
'la',  
'le',  
'les',  
'leur',  
'là',  
'ma',
```

```
'maintenant',  
'mais',  
'mes',  
'mine',  
'moins',  
'mon',  
'mot',  
'même',  
'ni',  
'nommés',  
'notre',  
'nous',  
'nouveaux',  
'ou',  
'où',  
'par',  
'parce',  
'parole',  
'pas',  
'personnes',  
'peut',  
'peu',  
'pièce',  
'plupart',  
'pour',  
'pourquoi',  
'quand',  
'que',  
'quel',  
'quelle',  
'quelles',  
'quels',  
'qui',  
'sa',  
'sans',  
'ses',  
'seulement',  
'si',  
'sien',  
'son',  
'sont',  
'sous',  
'soyez',  
'sujet',  
'sur',  
'ta',  
'tandis',  
'tellement',  
'tels',  
'tes',  
'ton',  
'tous',  
'tout',  
'trop',  
'très',  
'tu',  
'valeur',  
'voie',  
'voient',  
'vont',  
'votre',  
'vous',  
'vu',  
'ça',  
'étaient',  
'état',  
'étions',  
'été',  
'être'  
];
```


Ticket::SearchIndex::StopWords###it

Italian stop words for fulltext index. These words will be removed from the search index.

Standardwert:

```
$Self->{'Ticket::SearchIndex::StopWords'}->{'it'} = [  
'a',  
'adesso',  
'ai',  
'al',  
'alla',  
'allo',  
'allora',  
'altre',  
'altri',  
'altro',  
'anche',  
'ancora',  
'avere',  
'aveva',  
'avevano',  
'ben',  
'buono',  
'che',  
'chi',  
'cinque',  
'comprare',  
'con',  
'consecutivi',  
'consecutivo',  
'cosa',  
'cui',  
'da',  
'del',  
'della',  
'dello',  
'dentro',  
'deve',  
'devo',  
'di',  
'doppio',  
'due',  
'e',  
'ecco',  
'fare',  
'fine',  
'fino',  
'fra',  
'gente',  
'giu',  
'ha',  
'hai',  
'hanno',  
'ho',  
'il',  
'indietro',  
'invece',  
'io',  
'la',  
'lavoro',  
'le',  
'lei',  
'lo',  
'loro',  
'lui',  
'lungo',  
'ma',  
'me',  
'meglio',  
'molta',  
'molti',
```

```
'molto',  
'nei',  
'nella',  
'no',  
'noi',  
'nome',  
'nostro',  
'nove',  
'nuovi',  
'nuovo',  
'o',  
'oltre',  
'ora',  
'otto',  
'peggio',  
'pero',  
'persone',  
'piu',  
'poco',  
'primo',  
'promesso',  
'qua',  
'quarto',  
'quasi',  
'quattro',  
'quello',  
'questo',  
'qui',  
'quindi',  
'quinto',  
'rispetto',  
'sara',  
'secondo',  
'sei',  
'sembra',  
'sembrava',  
'senza',  
'sette',  
'sia',  
'siamo',  
'siete',  
'solo',  
'sono',  
'sopra',  
'soprattutto',  
'sotto',  
'stati',  
'stato',  
'stesso',  
'su',  
'subito',  
'sul',  
'sulla',  
'tanto',  
'te',  
'tempo',  
'terzo',  
'tra',  
'tre',  
'triplo',  
'ultimo',  
'un',  
'una',  
'uno',  
'va',  
'vai',  
'voi',  
'volte',  
'vostro'  
];
```

Ticket::SearchIndex::StopWords###Custom

Customizable stop words for fulltext index. These words will be removed from the search index.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::SearchIndex::StopWords'}->{'Custom'} = [
  'MyStopWord'
];
```

Ticket::EventModulePost###2000-ArticleSearchIndex

Builds an article index right after the article's creation.

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'2000-ArticleSearchIndex'} = {
  'Event' => '(ArticleCreate|ArticleUpdate)',
  'Module' => 'Kernel::System::Ticket::Event::ArticleSearchIndex'
};
```

Ticket → Core::LinkObject

LinkObject::PossibleLink###0200

Links 2 tickets with a "Normal" type link.

Standardwert:

```
$Self->{'LinkObject::PossibleLink'}->{'0200'} = {
  'Object1' => 'Ticket',
  'Object2' => 'Ticket',
  'Type' => 'Normal'
};
```

LinkObject::PossibleLink###0201

Links 2 tickets with a "ParentChild" type link.

Standardwert:

```
$Self->{'LinkObject::PossibleLink'}->{'0201'} = {
  'Object1' => 'Ticket',
  'Object2' => 'Ticket',
  'Type' => 'ParentChild'
};
```

LinkObject::IgnoreLinkedTicketStateTypes

Defines, which tickets of which ticket state types should not be listed in linked ticket lists.

Standardwert:

```
$Self->{'LinkObject::IgnoreLinkedTicketStateTypes'} = [
  'merged',
  'removed'
];
```

LinkObject::StrikeThroughLinkedTicketStateTypes

For these state types the ticket numbers are striked through in the link table.

Standardwert:

```
$Self->{'LinkObject::StrikeThroughLinkedTicketStateTypes'} = [
  'merged'
];
```

Ticket → Core::PostMaster

PostmasterMaxEmails

Maximal auto email responses to own email-address a day (Loop-Protection).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PostmasterMaxEmails'} = '40';
```

PostmasterMaxEmailsPerAddress

Maximal auto email responses to own email-address a day, configurable by email address (Loop-Protection).

Standardwert:

```
$Self->{'PostmasterMaxEmailsPerAddress'} = {};
```

PostMasterMaxEmailSize

Maximal size in KBytes for mails that can be fetched via POP3/POP3S/IMAP/IMAPS (KBytes).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PostMasterMaxEmailSize'} = '16384';
```

PostMasterReconnectMessage

The maximum number of mails fetched at once before reconnecting to the server.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PostMasterReconnectMessage'} = '20';
```

LoopProtectionModule

Default loop protection module.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'LoopProtectionModule'} = 'Kernel::System::PostMaster::LoopProtection::DB';
```

LoopProtectionLog

Path for the log file (it only applies if "FS" was selected for LoopProtectionModule and it is mandatory).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'LoopProtectionLog'} = '<OTRS_CONFIG_Home>/var/log/LoopProtection';
```

PostmasterAutoHTML2Text

Converts HTML mails into text messages.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PostmasterAutoHTML2Text'} = '1';
```

PostmasterUserID

Specifies user id of the postmaster data base.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PostmasterUserID'} = '1';
```

PostmasterDefaultQueue

Defines the postmaster default queue.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PostmasterDefaultQueue'} = 'Raw';
```

PostmasterDefaultPriority

Defines the default priority of new tickets.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PostmasterDefaultPriority'} = '3 normal';
```

PostmasterDefaultState

Defines the default state of new tickets.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PostmasterDefaultState'} = 'new';
```

PostmasterFollowUpState

Defines the state of a ticket if it gets a follow-up.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PostmasterFollowUpState'} = 'open';
```

PostmasterFollowUpStateClosed

Defines the state of a ticket if it gets a follow-up and the ticket was already closed.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PostmasterFollowUpStateClosed'} = 'open';
```

PostmasterFollowUpOnUnlockAgentNotifyOnlyToOwner

Sends agent follow-up notification only to the owner, if a ticket is unlocked (the default is to send the notification to all agents).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PostmasterFollowUpOnUnlockAgentNotifyOnlyToOwner'} = '0';
```

PostmasterHeaderFieldCount

Defines the number of header fields in frontend modules for add and update postmaster filters. It can be up to 99 fields.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PostmasterHeaderFieldCount'} = '12';
```

PostmasterX-Header

Defines all the X-headers that should be scanned.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PostmasterX-Header'} = [  
  'From',  
  'To',  
  'Cc',  
  'Reply-To',  
  'ReplyTo',  
  'Subject',  
  'Message-ID',  
  'Message-Id',  
  'Resent-To',  
  'Resent-From',  
  'Precedence',  
  'Mailing-List',  
  'List-Id',  
  'List-Archive',  
  'Errors-To',  
  'References',  
  'In-Reply-To',  
  'Auto-Submitted',  
  'X-Loop',  
  'X-Spam-Flag',  
  'X-Spam-Level',  
  'X-Spam-Score',  
  'X-Spam-Status',  
  'X-No-Loop',  
  'X-Priority',  
  'Importance',  
  'X-Mailer',  
  'User-Agent',  
  'Organization',  
  'X-Original-To',  
  'Delivered-To',  
  'Envelope-To',  
  'X-Envelope-To',  
  'Return-Path',  
  'X-OTRS-Owner',  
  'X-OTRS-OwnerID',  
  'X-OTRS-Responsible',  
  'X-OTRS-ResponsibleID',  
  'X-OTRS-Loop',  
  'X-OTRS-Priority',  
  'X-OTRS-Queue',  
  'X-OTRS-Lock',  
  'X-OTRS-Ignore',  
  'X-OTRS-State',  
  'X-OTRS-State-PendingTime',  
  'X-OTRS-Type',  
  'X-OTRS-Service',  
  'X-OTRS-SLA',  
  'X-OTRS-Title',  
  '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',
'X-OTRS-FollowUp-Title',
'X-OTRS-BodyDecrypted'
];
```

PostMaster::PreFilterModule###1-Match

Module to filter and manipulate incoming messages. Block/ignore all spam email with From: noreply@ address.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PostMaster::PreFilterModule'}->{'1-Match'} = {
  'Match' => {
    'From' => 'noreply@'
  },
  'Module' => 'Kernel::System::PostMaster::Filter::Match',
  'Set' => {
    'X-OTRS-Ignore' => 'yes'
  }
};
```

PostMaster::PreFilterModule###2-Match

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 =>.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PostMaster::PreFilterModule'}->{'2-Match'} = {
  'Match' => {
    'Subject' => 'SomeNumber:(\d\d\d\d)'
  },
  'Module' => 'Kernel::System::PostMaster::Filter::Match',
  'Set' => {
    'X-OTRS-DynamicField-TicketFreeKey1' => 'SomeNumber',
    'X-OTRS-DynamicField-TicketFreeText1' => '[***]'
  }
};
```

PostMaster::PreFilterModule###3-NewTicketReject

Blocks all the incoming emails that do not have a valid ticket number in subject with From: @example.com address.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PostMaster::PreFilterModule'}->{'3-NewTicketReject'} = {
  'Match' => {
    'From' => '@example.com'
  },
  'Module' => 'Kernel::System::PostMaster::Filter::NewTicketReject',
  'Set' => {
    'X-OTRS-Ignore' => 'yes'
  }
};
```

```
}  
};
```

PostMaster::PreFilterModule::NewTicketReject::Sender

Defines the sender for rejected emails.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PostMaster::PreFilterModule::NewTicketReject::Sender'} =  
'noreply@example.com';
```

PostMaster::PreFilterModule::NewTicketReject::Subject

Defines the subject for rejected emails.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PostMaster::PreFilterModule::NewTicketReject::Subject'} = 'Email Rejected';
```

PostMaster::PreFilterModule::NewTicketReject::Body

Defines the body text for rejected emails.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$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  
';
```

PostMaster::PreFilterModule###4-CMD

CMD example setup. Ignores emails where external CMD returns some output on STDOUT (email will be piped into STDIN of some.bin).

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PostMaster::PreFilterModule'}->{'4-CMD'} = {  
'CMD' => '/usr/bin/some.bin',  
'Module' => 'Kernel::System::PostMaster::Filter::CMD',  
'Set' => {  
  'X-OTRS-Ignore' => 'yes'  
}  
};
```

PostMaster::PreFilterModule###5-SpamAssassin

Spam Assassin example setup. Ignores emails that are marked with SpamAssassin.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PostMaster::PreFilterModule'}->{'5-SpamAssassin'} = {  
'CMD' => '/usr/bin/spamassassin | grep -i "X-Spam-Status: yes"',  
'Module' => 'Kernel::System::PostMaster::Filter::CMD',
```



```
'Set' => {
  'X-OTRS-Ignore' => 'yes'
}
};
```

PostMaster::PreFilterModule###6-SpamAssassin

Spam Assassin example setup. Moves marked mails to spam queue.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$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'
  }
};
```

PostMaster::PreFilterModule###000-MatchDBSource

Module to use database filter storage.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'PostMaster::PreFilterModule'}->{'000-MatchDBSource'} = {
  'Module' => 'Kernel::System::PostMaster::Filter::MatchDBSource'
};
```

PostMaster::PostFilterModule###000-FollowUpArticleTypeCheck

Module to check if arrived emails should be marked as email-internal (because of original forwarded internal email). ArticleType and SenderType define the values for the arrived email/article.

Standardwert:

```
$Self->{'PostMaster::PostFilterModule'}->{'000-FollowUpArticleTypeCheck'} = {
  'ArticleType' => 'email-internal',
  'Module' => 'Kernel::System::PostMaster::Filter::FollowUpArticleTypeCheck',
  'SenderType' => 'customer'
};
```

PostMaster::PreFilterModule###000-ExternalTicketNumberRecognition1

Recognize if a ticket is a follow-up to an existing ticket using an external ticket number.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PostMaster::PreFilterModule'}->{'000-ExternalTicketNumberRecognition1'} = {
  'ArticleType' => 'note-report',
  'DynamicFieldName' => 'Name_X',
  'FromAddressRegExp' => '\\s*@example.com',
  'Module' => 'Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition',
  'Name' => 'Some Description',
  'NumberRegExp' => '\\s*Incident-(\\d.*)\\s*',
  'SearchInBody' => '1',
  'SearchInSubject' => '1',
  'SenderType' => 'system',
  'TicketStateTypes' => 'new;open'
};
```

PostMaster::PreFilterModule###000-ExternalTicketNumberRecognition2

Recognize if a ticket is a follow-up to an existing ticket using an external ticket number.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PostMaster::PreFilterModule'}->{'000-ExternalTicketNumberRecognition2'} = {
  'ArticleType' => 'note-report',
  'DynamicFieldName' => 'Name_X',
  'FromAddressRegExp' => '\\s*@example.com',
  'Module' => 'Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition',
  'Name' => 'Some Description',
  'NumberRegExp' => '\\s*Incident-(\\d.*)\\s*',
  'SearchInBody' => '1',
  'SearchInSubject' => '1',
  'SenderType' => 'system',
  'TicketStateTypes' => 'new;open'
};
```

PostMaster::PreFilterModule###000-ExternalTicketNumberRecognition3

Recognize if a ticket is a follow-up to an existing ticket using an external ticket number.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PostMaster::PreFilterModule'}->{'000-ExternalTicketNumberRecognition3'} = {
  'ArticleType' => 'note-report',
  'DynamicFieldName' => 'Name_X',
  'FromAddressRegExp' => '\\s*@example.com',
  'Module' => 'Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition',
  'Name' => 'Some Description',
  'NumberRegExp' => '\\s*Incident-(\\d.*)\\s*',
  'SearchInBody' => '1',
  'SearchInSubject' => '1',
  'SenderType' => 'system',
  'TicketStateTypes' => 'new;open'
};
```

PostMaster::PreFilterModule###000-ExternalTicketNumberRecognition4

Recognize if a ticket is a follow-up to an existing ticket using an external ticket number.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PostMaster::PreFilterModule'}->{'000-ExternalTicketNumberRecognition4'} = {
  'ArticleType' => 'note-report',
  'DynamicFieldName' => 'Name_X',
  'FromAddressRegExp' => '\\s*@example.com',
  'Module' => 'Kernel::System::PostMaster::Filter::ExternalTicketNumberRecognition',
  'Name' => 'Some Description',
  'NumberRegExp' => '\\s*Incident-(\\d.*)\\s*',
  'SearchInBody' => '1',
  'SearchInSubject' => '1',
  'SenderType' => 'system',
  'TicketStateTypes' => 'new;open'
};
```

PostMaster::PreFilterModule###000-DecryptBody

Module to filter encrypted bodies of incoming messages.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PostMaster::PreFilterModule'}->{'000-DecryptBody'} = {
  'Module' => 'Kernel::System::PostMaster::Filter::Decrypt',
  'StoreDecryptedBody' => '0'
};
```

PostMaster::PreFilterModule###000-SMIMEFetchFromCustomer

Module to fetch customer users SMIME certificates of incoming messages.

Standardwert:

```
$Self->{'PostMaster::PreFilterModule'}->{'000-SMIMEFetchFromCustomer'} = {  
  'Module' => 'Kernel::System::PostMaster::Filter::SMIMEFetchFromCustomer'  
};
```

PostMaster::CheckFollowUpModule###0100-Subject

Checks if an E-Mail is a followup to an existing ticket by searching the subject for a valid ticket number.

Standardwert:

```
$Self->{'PostMaster::CheckFollowUpModule'}->{'0100-Subject'} = {  
  'Module' => 'Kernel::System::PostMaster::FollowUpCheck::Subject'  
};
```

PostMaster::CheckFollowUpModule###0200-References

Executes follow-up checks on In-Reply-To or References headers for mails that don't have a ticket number in the subject.

Standardwert:

```
$Self->{'PostMaster::CheckFollowUpModule'}->{'0200-References'} = {  
  'Module' => 'Kernel::System::PostMaster::FollowUpCheck::References'  
};
```

PostMaster::CheckFollowUpModule###0300-Body

Executes follow-up checks on email body for mails that don't have a ticket number in the subject.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PostMaster::CheckFollowUpModule'}->{'0300-Body'} = {  
  'Module' => 'Kernel::System::PostMaster::FollowUpCheck::Body'  
};
```

PostMaster::CheckFollowUpModule###0400-Attachments

Executes follow-up checks on attachment contents for mails that don't have a ticket number in the subject.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PostMaster::CheckFollowUpModule'}->{'0400-Attachments'} = {  
  'Module' => 'Kernel::System::PostMaster::FollowUpCheck::Attachments'  
};
```

PostMaster::CheckFollowUpModule###0500-RawEmail

Executes follow-up checks on the raw source email for mails that don't have a ticket number in the subject.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'PostMaster::CheckFollowUpModule'}->{'0500-RawEmail'} = {  
  'Module' => 'Kernel::System::PostMaster::FollowUpCheck::RawEmail'  
};
```

SendNoAutoResponseRegExp

If this regex matches, no message will be send by the autoresponder.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'SendNoAutoResponseRegExp'} = '(MAILER-DAEMON|postmaster|abuse)@.+?\.\.?';
```

AutoResponseForWebTickets

If this option is set to 'Yes', tickets created via the web interface, via Customers or Agents, will receive an autoresponse if configured. If this option is set to 'No', no auto-responses will be sent.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'AutoResponseForWebTickets'} = '1';
```

Ticket → Core::Queue

Queue::EventModulePost###2300-UpdateQueue

Event module that performs an update statement on TicketIndex to rename the queue name there if needed and if StaticDB is actually used.

Standardwert:

```
$Self->{'Queue::EventModulePost'}->{'2300-UpdateQueue'} = {
  'Event' => 'QueueUpdate',
  'Module' => 'Kernel::System::Queue::Event::TicketAcceleratorUpdate',
  'Transaction' => '0'
};
```

Ticket → Core::Stats

Stats::DynamicObjectRegistration###Ticket

Module to generate ticket statistics.

Standardwert:

```
$Self->{'Stats::DynamicObjectRegistration'}->{'Ticket'} = {
  'Module' => 'Kernel::System::Stats::Dynamic::Ticket'
};
```

Stats::DynamicObjectRegistration###TicketList

Determines if the statistics module may generate ticket lists.

Standardwert:

```
$Self->{'Stats::DynamicObjectRegistration'}->{'TicketList'} = {
  'Module' => 'Kernel::System::Stats::Dynamic::TicketList'
};
```

Stats::DynamicObjectRegistration###TicketAccountedTime

Module to generate accounted time ticket statistics.

Standardwert:

```
$Self->{'Stats::DynamicObjectRegistration'}->{'TicketAccountedTime'} = {
  'Module' => 'Kernel::System::Stats::Dynamic::TicketAccountedTime'
};
```

Stats::DynamicObjectRegistration###TicketSolutionResponseTime

Module to generate ticket solution and response time statistics.

Standardwert:

```
$Self->{'Stats::DynamicObjectRegistration'}->{'TicketSolutionResponseTime'} = {
```

```
'Module' => 'Kernel::System::Stats::Dynamic::TicketSolutionResponseTime'  
};
```

Ticket → Core::Ticket

Ticket::Hook

The identifier for a ticket, e.g. Ticket#, Call#, MyTicket#. The default is Ticket#.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Hook'} = 'Ticket#';
```

Ticket::HookDivider

The divider between TicketHook and ticket number. E.g ': '.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::HookDivider'} = '';
```

Ticket::SubjectSize

Max size of the subjects in an email reply and in some overview screens.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::SubjectSize'} = '100';
```

Ticket::SubjectRe

The text at the beginning of the subject in an email reply, e.g. RE, AW, or AS.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::SubjectRe'} = 'Re';
```

Ticket::SubjectFwd

The text at the beginning of the subject when an email is forwarded, e.g. FW, Fwd, or WG.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::SubjectFwd'} = 'Fwd';
```

Ticket::SubjectFormat

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 latter case you should verify that the setting PostMaster::CheckFollowUpModule###0200-References is activated to recognize followups based on email headers.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::SubjectFormat'} = 'Left';
```

Ticket::MergeDynamicFields

A list of dynamic fields that are merged into the main ticket during a merge operation. Only dynamic fields that are empty in the main ticket will be set.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::MergeDynamicFields'} = [];
```

Ticket::CustomQueue

Name of custom queue. The custom queue is a queue selection of your preferred queues and can be selected in the preferences settings.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::CustomQueue'} = 'My Queues';
```

Ticket::CustomService

Name of custom service. The custom service is a service selection of your preferred services and can be selected in the preferences settings.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::CustomService'} = 'My Services';
```

Ticket::NewArticleIgnoreSystemSender

Ignore article with system sender type for new article feature (e. g. auto responses or email notifications).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::NewArticleIgnoreSystemSender'} = '0';
```

Ticket::ChangeOwnerToEveryone

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::ChangeOwnerToEveryone'} = '0';
```

Ticket::Responsible

Enables ticket responsible feature, to keep track of a specific ticket.

Standardwert:

```
$Self->{'Ticket::Responsible'} = '0';
```

Ticket::ResponsibleAutoSet

Automatically sets the owner of a ticket as the responsible for it (if ticket responsible feature is enabled). This will only work by manually actions of the logged in user. It does not work for automated actions e.g. GenericAgent, Postmaster and GenericInterface.

Standardwert:

```
$Self->{'Ticket::ResponsibleAutoSet'} = '1';
```

Ticket::InvalidOwner::StateChange

Automatically change the state of a ticket with an invalid owner once it is unlocked.
Maps from a state type to a new ticket state.

Standardwert:

```
$Self->{'Ticket::InvalidOwner::StateChange'} = {  
  'pending auto' => 'open',  
  'pending reminder' => 'open'  
};
```

Ticket::Type

Allows defining new types for ticket (if ticket type feature is enabled).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Type'} = '0';
```

Ticket::Type::Default

Defines the default ticket type.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Type::Default'} = 'Unclassified';
```

Ticket::Service

Allows defining services and SLAs for tickets (e. g. email, desktop, network, ...), and escalation attributes for SLAs (if ticket service/SLA feature is enabled).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Service'} = '0';
```

Ticket::Service::KeepChildren

Retains all services in listings even if they are children of invalid elements.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Service::KeepChildren'} = '0';
```

Ticket::Service::Default::UnknownCustomer

Allows default services to be selected also for non existing customers.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Service::Default::UnknownCustomer'} = '0';
```

Ticket::ArchiveSystem

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::ArchiveSystem'} = '0';
```

Ticket::ArchiveSystem::RemoveSeenFlags

Controls if the ticket and article seen flags are removed when a ticket is archived.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::ArchiveSystem::RemoveSeenFlags'} = '1';
```

Ticket::ArchiveSystem::RemoveTicketWatchers

Removes the ticket watcher information when a ticket is archived.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::ArchiveSystem::RemoveTicketWatchers'} = '1';
```

Ticket::CustomerArchiveSystem

Activates the ticket archive system search in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::CustomerArchiveSystem'} = '0';
```

Ticket::NumberGenerator

Selects the ticket number generator module. "AutoIncrement" increments the ticket number, the SystemID and the counter are used with SystemID.counter format (e.g. 1010138, 1010139). With "Date" the ticket numbers will be generated by the current date, the SystemID and the counter. The format looks like Year.Month.Day.SystemID.counter (e.g. 200206231010138, 200206231010139). With "DateChecksum" the counter will be appended as checksum to the string of date and SystemID. The checksum will be rotated on a daily basis. The format looks like Year.Month.Day.SystemID.Counter.CheckSum (e.g. 2002070110101520, 2002070110101535). "Random" generates randomized ticket numbers in the format "SystemID.Random" (e.g. 100057866352, 103745394596).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::NumberGenerator'} = 'Kernel::System::Ticket::Number::DateChecksum';
```

Ticket::NumberGenerator::CheckSystemID

Checks the SystemID in ticket number detection for follow-ups (use "No" if SystemID has been changed after using the system).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::NumberGenerator::CheckSystemID'} = '1';
```

Ticket::NumberGenerator::MinCounterSize

Sets the minimal ticket counter size if "AutoIncrement" was selected as TicketNumberGenerator. Default is 5, this means the counter starts from 10000.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:


```
$Self->{'Ticket::NumberGenerator::MinCounterSize'} = '5';
```

Ticket::NumberGenerator::Date::UseFormattedCounter

Enables the minimal ticket counter size (if "Date" was selected as TicketNumberGenerator).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::NumberGenerator::Date::UseFormattedCounter'} = '0';
```

Ticket::CounterLog

Log file for the ticket counter.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::CounterLog'} = '<OTRS_CONFIG_Home>/var/log/TicketCounter.log';
```

Ticket::IndexModule

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 command "bin/otrs.Console.pl Maint::Ticket::QueueIndexRebuild" for initial index creation.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::IndexModule'} =  
'Kernel::System::Ticket::IndexAccelerator::RuntimeDB';
```

Ticket::StorageModule

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. Note: Searching for attachment names is not supported when "FS" is used.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

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

Ticket::StorageModule::CheckAllBackends

Specifies whether all storage backends should be checked when looking for attachments. This is only required for installations where some attachments are in the file system, and others in the database.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::StorageModule::CheckAllBackends'} = '0';
```

ArticleDir

Specifies the directory to store the data in, if "FS" was selected for TicketStorageModule.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'ArticleDir'} = '<OTRS_CONFIG_Home>/var/article';
```

OTRSEscalationEvents::DecayTime

The duration in minutes after emitting an event, in which the new escalation notify and start events are suppressed.

Standardwert:

```
$Self->{'OTRSEscalationEvents::DecayTime'} = '1440';
```

Ticket::EventModulePost###2300-ArchiveRestore

Restores a ticket from the archive (only if the event is a state change to any open available state).

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'2300-ArchiveRestore'} = {
  'Event' => 'TicketStateUpdate',
  'Module' => 'Kernel::System::Ticket::Event::ArchiveRestore'
};
```

Ticket::EventModulePost###2600-AcceleratorUpdate

Updates the ticket index accelerator.

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'2600-AcceleratorUpdate'} = {
  'Event' => 'TicketStateUpdate|TicketQueueUpdate|TicketLockUpdate',
  'Module' => 'Kernel::System::Ticket::Event::TicketAcceleratorUpdate'
};
```

Ticket::EventModulePost###2700-ForceOwnerResetOnMove

Resets and unlocks the owner of a ticket if it was moved to another queue.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'2700-ForceOwnerResetOnMove'} = {
  'Event' => 'TicketQueueUpdate',
  'Module' => 'Kernel::System::Ticket::Event::ForceOwnerReset'
};
```

Ticket::EventModulePost###2800-ForceStateChangeOnLock

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'2800-ForceStateChangeOnLock'} = {
  'Event' => 'TicketLockUpdate',
  'Module' => 'Kernel::System::Ticket::Event::ForceState',
  'new' => 'open'
};
```

Ticket::EventModulePost###3000-ResponsibleAutoSet

Automatically sets the responsible of a ticket (if it is not set yet) after the first owner update.

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'3000-ResponsibleAutoSet'} = {
  'Event' => 'TicketOwnerUpdate',
  'Module' => 'Kernel::System::Ticket::Event::ResponsibleAutoSet'
};
```

Ticket::EventModulePost###3300-TicketPendingTimeReset

Sets the PendingTime of a ticket to 0 if the state is changed to a non-pending state.

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'3300-TicketPendingTimeReset'} = {
  'Event' => 'TicketStateUpdate',
  'Module' => 'Kernel::System::Ticket::Event::TicketPendingTimeReset'
};
```

Ticket::EventModulePost###7000-NotificationEvent

Sends the notifications which are configured in the admin interface under "Notification (Event)".

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'7000-NotificationEvent'} = {
  'Event' => '',
  'Module' => 'Kernel::System::Ticket::Event::NotificationEvent',
  'Transaction' => '1'
};
```

Ticket::EventModulePost###6000-EscalationIndex

Updates the ticket escalation index after a ticket attribute got updated.

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'6000-EscalationIndex'} = {
  'Event' => 'TicketSLAUpdate|TicketQueueUpdate|TicketStateUpdate|TicketCreate|
ArticleCreate|TicketDynamicFieldUpdate|TicketTypeUpdate|TicketServiceUpdate|
TicketCustomerUpdate|TicketPriorityUpdate|TicketMerge',
  'Module' => 'Kernel::System::Ticket::Event::TicketEscalationIndex',
  'Transaction' => '1'
};
```

Ticket::EventModulePost###4300-EscalationStopEvents

Ticket event module that triggers the escalation stop events.

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'4300-EscalationStopEvents'} = {
  'Event' => 'TicketSLAUpdate|TicketQueueUpdate|TicketStateUpdate|ArticleCreate',
  'Module' => 'Kernel::System::Ticket::Event::TriggerEscalationStopEvents',
  'Transaction' => '0'
};
```

Ticket::EventModulePost###3600-ForceUnlockOnMove

Forces to unlock tickets after being moved to another queue.

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'3600-ForceUnlockOnMove'} = {
  'Event' => 'TicketQueueUpdate',
  'Module' => 'Kernel::System::Ticket::Event::ForceUnlock'
};
```

Ticket::EventModulePost###4000-TicketArticleNewMessageUpdate

Update Ticket "Seen" flag if every article got seen or a new Article got created.

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'4000-TicketArticleNewMessageUpdate'} = {
  'Event' => 'ArticleCreate|ArticleFlagSet',
  'Module' => 'Kernel::System::Ticket::Event::TicketNewMessageUpdate'
};
```

```
};
```

DynamicFieldFromCustomerUser::Mapping

Define a mapping between variables of the customer user data (keys) and dynamic fields of a ticket (values). The purpose is to store customer user data in ticket dynamic fields. The dynamic fields must be present in the system and should be enabled for AgentTicketFreeText, so that they can be set/updated manually by the agent. They mustn't be enabled for AgentTicketPhone, AgentTicketEmail and AgentTicketCustomer. If they were, they would have precedence over the automatically set values. To use this mapping, you have to also activate the next setting below.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'DynamicFieldFromCustomerUser::Mapping'} = {  
  'UserFirstname' => 'CustomerFirstname'  
};
```

Ticket::EventModulePost###4100-DynamicFieldFromCustomerUser

This event module stores attributes from CustomerUser as DynamicFields tickets. Please see the setting above for how to configure the mapping.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'4100-DynamicFieldFromCustomerUser'} = {  
  'Event' => '(TicketCreate|TicketCustomerUpdate)',  
  'Module' => 'Kernel::System::Ticket::Event::DynamicFieldFromCustomerUser'  
};
```

Ticket::CustomModule###001-CustomModule

Overloads (redefines) existing functions in Kernel::System::Ticket. Used to easily add customizations.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::CustomModule'}->{'001-CustomModule'} =  
'Kernel::System::Ticket::Custom';
```

Ticket::ViewableSenderTypes

Defines the default viewable sender types of a ticket (default: customer).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::ViewableSenderTypes'} = [  
  '\customer\  
];
```

Ticket::ViewableLocks

Defines the viewable locks of a ticket. NOTE: When you change this setting, make sure to delete the cache in order to use the new value. Default: unlock, tmp_lock.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::ViewableLocks'} = [  
  '\unlock\  
  '\tmp_lock\  
];
```

Ticket::ViewableStateType

Defines the valid state types for a ticket.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::ViewableStateType'} = [  
  'new',  
  'open',  
  'pending reminder',  
  'pending auto'  
];
```

Ticket::UnlockStateType

Defines the valid states for unlocked tickets. To unlock tickets the script "bin/otrs.Console.pl Maint::Ticket::UnlockTimeout" can be used.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::UnlockStateType'} = [  
  'new',  
  'open'  
];
```

Ticket::PendingNotificationOnlyToOwner

Sends reminder notifications of unlocked ticket after reaching the reminder date (only sent to ticket owner).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::PendingNotificationOnlyToOwner'} = '0';
```

Ticket::PendingNotificationNotToResponsible

Disables sending reminder notifications to the responsible agent of a ticket (Ticket::Responsible needs to be activated).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::PendingNotificationNotToResponsible'} = '0';
```

Ticket::PendingReminderStateType

Defines the state type of the reminder for pending tickets.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::PendingReminderStateType'} = [  
  'pending reminder'  
];
```

Ticket::PendingAutoStateType

Determines the possible states for pending tickets that changed state after reaching time limit.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::PendingAutoStateType'} = [  
  'pending auto'  
];
```

Ticket::StateAfterPending

Defines which states should be set automatically (Content), after the pending time of state (Key) has been reached.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::StateAfterPending'} = {  
  'pending auto close+' => 'closed successful',  
  'pending auto close-' => 'closed unsuccessful'  
};
```

System::Permission

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'System::Permission'} = [  
  'ro',  
  'move_into',  
  'create',  
  'note',  
  'owner',  
  'priority',  
  'rw'  
];
```

Ticket::Permission###1-OwnerCheck

Module to grant access to the owner of a ticket.

Standardwert:

```
$Self->{'Ticket::Permission'}->{'1-OwnerCheck'} = {  
  'Granted' => '1',  
  'Module' => 'Kernel::System::Ticket::Permission::OwnerCheck',  
  'Required' => '0'  
};
```

Ticket::Permission::OwnerCheck::Queues

Optional queue limitation for the OwnerCheck permission module. If set, permission is only granted for tickets in the specified queues.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Permission::OwnerCheck::Queues'} = {  
  'Misc' => 'note',  
  'Postmaster' => 'ro, move, note',  
  'Raw' => 'rw'  
};
```

Ticket::Permission###2-ResponsibleCheck

Module to grant access to the agent responsible of a ticket.

Standardwert:

```
$Self->{'Ticket::Permission'}->{'2-ResponsibleCheck'} = {  
  'Granted' => '1',  
  'Module' => 'Kernel::System::Ticket::Permission::ResponsibleCheck',  
  'Required' => '0'  
};
```

Ticket::Permission::ResponsibleCheck::Queues

Optional queue limitation for the ResponsibleCheck permission module. If set, permission is only granted for tickets in the specified queues.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Permission::ResponsibleCheck::Queues'} = {  
  'Misc' => 'note',  
  'Postmaster' => 'ro, move, note',  
  'Raw' => 'rw'  
};
```

Ticket::Permission###3-GroupCheck

Module to check the group permissions for the access to tickets.

Standardwert:

```
$Self->{'Ticket::Permission'}->{'3-GroupCheck'} = {  
  'Granted' => '1',  
  'Module' => 'Kernel::System::Ticket::Permission::GroupCheck',  
  'Required' => '0'  
};
```

Ticket::Permission###4-WatcherCheck

Module to grant access to the watcher agents of a ticket.

Standardwert:

```
$Self->{'Ticket::Permission'}->{'4-WatcherCheck'} = {  
  'Granted' => '1',  
  'Module' => 'Kernel::System::Ticket::Permission::WatcherCheck',  
  'Required' => '0'  
};
```

Ticket::Permission###5-CreatorCheck

Module to grant access to the creator of a ticket.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Permission'}->{'5-CreatorCheck'} = {  
  'Granted' => '1',  
  'Module' => 'Kernel::System::Ticket::Permission::CreatorCheck',  
  'Required' => '0'  
};
```

Ticket::Permission::CreatorCheck::Queues

Optional queue limitation for the CreatorCheck permission module. If set, permission is only granted for tickets in the specified queues.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Permission::CreatorCheck::Queues'} = {  
  'Misc' => 'note',  
  'Postmaster' => 'ro, move, note',  
  'Raw' => 'rw'  
};
```

Ticket::Permission###6-InvolvedCheck

Module to grant access to any agent that has been involved in a ticket in the past (based on ticket history entries).

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Permission'}->{'6-InvolvedCheck'} = {  
  'Granted' => '1',  
  'Module' => 'Kernel::System::Ticket::Permission::InvolvedCheck',  
  'Required' => '0'  
};
```

Ticket::Permission::InvolvedCheck::Queues

Optional queue limitation for the InvolvedCheck permission module. If set, permission is only granted for tickets in the specified queues.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Permission::InvolvedCheck::Queues'} = {  
  'Misc' => 'note',  
  'Postmaster' => 'ro, move, note',  
  'Raw' => 'rw'  
};
```

CustomerTicket::Permission###1-GroupCheck

Module to check the group permissions for customer access to tickets.

Standardwert:

```
$Self->{'CustomerTicket::Permission'}->{'1-GroupCheck'} = {  
  'Granted' => '0',  
  'Module' => 'Kernel::System::Ticket::CustomerPermission::GroupCheck',  
  'Required' => '1'  
};
```

CustomerTicket::Permission###2-CustomerUserIDCheck

Module to grant access if the CustomerUserID of the ticket matches the CustomerUserID of the customer.

Standardwert:

```
$Self->{'CustomerTicket::Permission'}->{'2-CustomerUserIDCheck'} = {  
  'Granted' => '1',  
  'Module' => 'Kernel::System::Ticket::CustomerPermission::CustomerUserIDCheck',  
  'Required' => '0'  
};
```

CustomerTicket::Permission###3-CustomerIDCheck

Module to grant access if the CustomerID of the ticket matches the CustomerID of the customer.

Standardwert:

```
$Self->{'CustomerTicket::Permission'}->{'3-CustomerIDCheck'} = {  
  'Granted' => '1',  
  'Module' => 'Kernel::System::Ticket::CustomerPermission::CustomerIDCheck',  
  'Required' => '0'  
};
```

Ticket::DefineEmailFrom

Defines how the From field from the emails (sent from answers and email tickets) should look like.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::DefineEmailFrom'} = 'SystemAddressName';
```

Ticket::DefineEmailFromSeparator

Defines the separator between the agents real name and the given queue email address.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::DefineEmailFromSeparator'} = 'via';
```

CustomerNotifyJustToRealCustomer

Sends customer notifications just to the mapped customer.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerNotifyJustToRealCustomer'} = '0';
```

AgentSelfNotifyOnAction

Specifies if an agent should receive email notification of his own actions.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'AgentSelfNotifyOnAction'} = '0';
```

Ticket::EventModulePost###9700-GenericAgent

Event module registration. For more performance you can define a trigger event (e. g. Event => TicketCreate).

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'9700-GenericAgent'} = {  
  'Event' => '',  
  'Module' => 'Kernel::System::Ticket::Event::GenericAgent',  
  'Transaction' => '1'  
};
```

Ticket::GenericAgentTicketSearch###ExtendedSearchCondition

Allows extended search conditions in ticket search of the generic agent interface. With this feature you can search e. g. ticket title with this kind of conditions like "(*key1*&&*key2*)" or "(*key1*||*key2*)".

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::GenericAgentTicketSearch'}->{'ExtendedSearchCondition'} = '1';
```

Ticket::GenericAgentRunLimit

Set the limit of tickets that will be executed on a single genericagent job execution.

Standardwert:

```
$Self->{'Ticket::GenericAgentRunLimit'} = '4000';
```

Ticket::UnlockOnAway

Unlock tickets whenever a note is added and the owner is out of office.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::UnlockOnAway'} = '1';
```

Ticket::IncludeUnknownTicketCustomers

Include unknown customers in ticket filter.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::IncludeUnknownTicketCustomers'} = '0';
```

StandardTemplate::Types

Defines the list of types for templates.

Standardwert:

```
$Self->{'StandardTemplate::Types'} = {
  'Answer' => 'Answer',
  'Create' => 'Create',
  'Email' => 'Email',
  'Forward' => 'Forward',
  'Note' => 'Note',
  'PhoneCall' => 'Phone call'
};
```

StandardTemplate2QueueByCreating

List of default Standard Templates which are assigned automatically to new Queues upon creation.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'StandardTemplate2QueueByCreating'} = [
  ''
];
```

Ticket::Frontend::DefaultRecipientDisplayType

Default display type for recipient (To,Cc) names in AgentTicketZoom and CustomerTicketZoom.

Standardwert:

```
$Self->{'Ticket::Frontend::DefaultRecipientDisplayType'} = 'Realname';
```

Ticket::Frontend::DefaultSenderDisplayType

Default display type for sender (From) names in AgentTicketZoom and CustomerTicketZoom.

Standardwert:

```
$Self->{'Ticket::Frontend::DefaultSenderDisplayType'} = 'Realname';
```

Ticket → Core::TicketACL

Ticket::Acl::Module###1-Ticket::Acl::Module

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).

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Acl::Module'}->{'1-Ticket::Acl::Module'} = {  
  'Module' => 'Kernel::System::Ticket::Acl::CloseParentAfterClosedChilds',  
  'State' => [  
    'closed successful',  
    'closed unsuccessful'  
  ]  
};
```

TicketACL::Default::Action

Default ACL values for ticket actions.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TicketACL::Default::Action'} = {};
```

ACLKeysLevel1Match

Defines which items are available in first level of the ACL structure.

Standardwert:

```
$Self->{'ACLKeysLevel1Match'} = {  
  'Properties' => 'Properties',  
  'PropertiesDatabase' => 'PropertiesDatabase'  
};
```

ACLKeysLevel1Change

Defines which items are available in first level of the ACL structure.

Standardwert:

```
$Self->{'ACLKeysLevel1Change'} = {  
  'Possible' => 'Possible',  
  'PossibleAdd' => 'PossibleAdd',  
  'PossibleNot' => 'PossibleNot'  
};
```

ACLKeysLevel2::Possible

Defines which items are available in second level of the ACL structure.

Standardwert:

```
$Self->{'ACLKeysLevel2::Possible'} = {  
  'Action' => 'Action',  
  'ActivityDialog' => 'ActivityDialog',  
  'Process' => 'Process',  
  'Ticket' => 'Ticket'  
};
```

ACLKeysLevel2::PossibleAdd

Defines which items are available in second level of the ACL structure.

Standardwert:

```
$Self->{'ACLKeysLevel2::PossibleAdd'} = {  
  'Action' => 'Action',  
  'ActivityDialog' => 'ActivityDialog',  
  'Process' => 'Process',  
  'Ticket' => 'Ticket'  
};
```

ACLKeysLevel2::PossibleNot

Defines which items are available in second level of the ACL structure.

Standardwert:

```
$Self->{'ACLKeysLevel2::PossibleNot'} = {
  'Action' => 'Action',
  'ActivityDialog' => 'ActivityDialog',
  'Process' => 'Process',
  'Ticket' => 'Ticket'
};
```

ACLKeysLevel2::Properties

Defines which items are available in second level of the ACL structure.

Standardwert:

```
$Self->{'ACLKeysLevel2::Properties'} = {
  'CustomerUser' => 'CustomerUser',
  'DynamicField' => 'DynamicField',
  'Frontend' => 'Frontend',
  'Owner' => 'Owner',
  'Priority' => 'Priority',
  'Process' => 'Process',
  'Queue' => 'Queue',
  'Responsible' => 'Responsible',
  'SLA' => 'SLA',
  'Service' => 'Service',
  'State' => 'State',
  'Ticket' => 'Ticket',
  'Type' => 'Type',
  'User' => 'User'
};
```

ACLKeysLevel2::PropertiesDatabase

Defines which items are available in second level of the ACL structure.

Standardwert:

```
$Self->{'ACLKeysLevel2::PropertiesDatabase'} = {
  'CustomerUser' => 'CustomerUser',
  'DynamicField' => 'DynamicField',
  'Owner' => 'Owner',
  'Priority' => 'Priority',
  'Process' => 'Process',
  'Queue' => 'Queue',
  'Responsible' => 'Responsible',
  'SLA' => 'SLA',
  'Service' => 'Service',
  'State' => 'State',
  'Ticket' => 'Ticket',
  'Type' => 'Type',
  'User' => 'User'
};
```

ACLKeysLevel3::Actions###100-Default

Defines which items are available for 'Action' in third level of the ACL structure.

Standardwert:

```
$Self->{'ACLKeysLevel3::Actions'}->{'100-Default'} = [
  'AgentTicketBounce',
  'AgentTicketClose',
  'AgentTicketCompose',
  'AgentTicketCustomer',
  'AgentTicketForward',
  'AgentTicketEmailOutbound',
  'AgentTicketFreeText',
  'AgentTicketHistory',
  'AgentTicketLink',
  'AgentTicketLock',
  'AgentTicketMerge',
  'AgentTicketMove',
  'AgentTicketNote',
  'AgentTicketOwner',
```

```
'AgentTicketPending',
'AgentTicketPhone',
'AgentTicketPhoneInbound',
'AgentTicketPhoneOutbound',
'AgentTicketPlain',
'AgentTicketPrint',
'AgentTicketPriority',
'AgentTicketProcess',
'AgentTicketResponsible',
'AgentTicketSearch',
'AgentTicketWatcher',
'AgentTicketZoom',
'AgentLinkObject',
'CustomerTicketPrint',
'CustomerTicketProcess',
'CustomerTicketZoom'
];
```

ACL::CacheTTL

Cache time in seconds for the DB ACL backend.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'ACL::CacheTTL'} = '3600';
```

TicketACL::Debug::Enabled

If enabled debugging information for ACLs is logged.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'TicketACL::Debug::Enabled'} = '0';
```

TicketACL::Debug::LogPriority

Defines the priority in which the information is logged and presented.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'TicketACL::Debug::LogPriority'} = 'debug';
```

TicketACL::Debug::Filter###00-Default

Filter for debugging ACLs. Note: More ticket attributes can be added in the format <OTRS_TICKET_Attribute> e.g. <OTRS_TICKET_Priority>.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'TicketACL::Debug::Filter'}->{'00-Default'} = {
  '<OTRS_TICKET_TicketNumber>' => '',
  'ACLName' => ''
};
```

Ticket → Core::TicketBulkAction

Ticket::Frontend::BulkFeature

Enables ticket bulk action feature for the agent frontend to work on more than one ticket at a time.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::BulkFeature'} = '1';
```

Ticket::Frontend::BulkFeatureGroup

Enables ticket bulk action feature only for the listed groups.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::BulkFeatureGroup'} = [
  'admin',
  'users'
];
```

Ticket → Core::TicketDynamicFieldDefault

Ticket::EventModulePost###9600-TicketDynamicFieldDefault

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::EventModulePost'}->{'9600-TicketDynamicFieldDefault'} = {
  'Module' => 'Kernel::System::Ticket::Event::TicketDynamicFieldDefault',
  'Transaction' => '1'
};
```

Ticket::TicketDynamicFieldDefault###Element1

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (<http://otrs.github.io/doc/>), chapter "Ticket Event Module".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element1'} = {
  'Event' => 'TicketCreate',
  'Name' => 'Field1',
  'Value' => 'Default'
};
```

Ticket::TicketDynamicFieldDefault###Element2

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (<http://otrs.github.io/doc/>), chapter "Ticket Event Module".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element2'} = {
  'Event' => '',
  'Name' => '',
  'Value' => ''
};
```

Ticket::TicketDynamicFieldDefault###Element3

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the

trigger event. Please check the developer manual (<http://otrs.github.io/doc/>), chapter "Ticket Event Module".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element3'} = {  
  'Event' => '',  
  'Name' => '',  
  'Value' => ''  
};
```

Ticket::TicketDynamicFieldDefault###Element4

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (<http://otrs.github.io/doc/>), chapter "Ticket Event Module".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element4'} = {  
  'Event' => '',  
  'Name' => '',  
  'Value' => ''  
};
```

Ticket::TicketDynamicFieldDefault###Element5

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (<http://otrs.github.io/doc/>), chapter "Ticket Event Module".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element5'} = {  
  'Event' => '',  
  'Name' => '',  
  'Value' => ''  
};
```

Ticket::TicketDynamicFieldDefault###Element6

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (<http://otrs.github.io/doc/>), chapter "Ticket Event Module".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element6'} = {  
  'Event' => '',  
  'Name' => '',  
  'Value' => ''  
};
```

Ticket::TicketDynamicFieldDefault###Element7

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (<http://otrs.github.io/doc/>), chapter "Ticket Event Module".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element7'} = {  
  'Event' => '',  
  'Name' => '',  
  'Value' => ''  
};
```

Ticket::TicketDynamicFieldDefault###Element8

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (<http://otrs.github.io/doc/>), chapter "Ticket Event Module".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element8'} = {  
  'Event' => '',  
  'Name' => '',  
  'Value' => ''  
};
```

Ticket::TicketDynamicFieldDefault###Element9

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (<http://otrs.github.io/doc/>), chapter "Ticket Event Module".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element9'} = {  
  'Event' => '',  
  'Name' => '',  
  'Value' => ''  
};
```

Ticket::TicketDynamicFieldDefault###Element10

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (<http://otrs.github.io/doc/>), chapter "Ticket Event Module".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element10'} = {  
  'Event' => '',  
  'Name' => '',  
  'Value' => ''  
};
```

Ticket::TicketDynamicFieldDefault###Element11

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (<http://otrs.github.io/doc/>), chapter "Ticket Event Module".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element11'} = {  
  'Event' => '',  
  'Name' => '',  
  'Value' => ''  
};
```

Ticket::TicketDynamicFieldDefault###Element12

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (<http://otrs.github.io/doc/>), chapter "Ticket Event Module".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element12'} = {  
  'Event' => '',  
  'Name' => '',  
  'Value' => ''  
};
```

Ticket::TicketDynamicFieldDefault###Element13

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (<http://otrs.github.io/doc/>), chapter "Ticket Event Module".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element13'} = {  
  'Event' => '',  
  'Name' => '',  
  'Value' => ''  
};
```

Ticket::TicketDynamicFieldDefault###Element14

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (<http://otrs.github.io/doc/>), chapter "Ticket Event Module".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element14'} = {  
  'Event' => '',  
  'Name' => '',  
  'Value' => ''  
};
```

Ticket::TicketDynamicFieldDefault###Element15

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (<http://otrs.github.io/doc/>), chapter "Ticket Event Module".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element15'} = {  
  'Event' => '',  
  'Name' => '',  
  'Value' => ''  
};
```

Ticket::TicketDynamicFieldDefault###Element16

Configures a default TicketDynamicField setting. "Name" defines the dynamic field which should be used, "Value" is the data that will be set, and "Event" defines the trigger event. Please check the developer manual (<http://otrs.github.io/doc/>), chapter "Ticket Event Module".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::TicketDynamicFieldDefault'}->{'Element16'} = {  
  'Event' => '',  
  'Name' => '',  
  'Value' => ''  
};
```

Ticket → Core::TicketWatcher

Ticket::Watcher

Enables or disables the ticket watcher feature, to keep track of tickets without being the owner nor the responsible.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Watcher'} = '0';
```

Ticket::WatcherGroup

Enables ticket watcher feature only for the listed groups.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::WatcherGroup'} = [  
  'admin',  
  'users'  
];
```

Ticket → Frontend::Admin

Events###Ticket

List of all ticket events to be displayed in the GUI.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Events'}->{'Ticket'} = [  
  'TicketCreate',  
  'TicketDelete',  
  'TicketTitleUpdate',  
  'TicketUnlockTimeoutUpdate',  
  'TicketQueueUpdate',  
  'TicketTypeUpdate',  
  'TicketServiceUpdate',  
  'TicketSLAUpdate',  
  'TicketCustomerUpdate',  
  'TicketPendingTimeUpdate',  
];
```

```
'TicketLockUpdate',
'TicketArchiveFlagUpdate',
'TicketStateUpdate',
'TicketOwnerUpdate',
'TicketResponsibleUpdate',
'TicketPriorityUpdate',
'HistoryAdd',
'HistoryDelete',
'TicketAccountTime',
'TicketMerge',
'TicketSubscribe',
'TicketUnsubscribe',
'TicketFlagSet',
'TicketFlagDelete',
'TicketSlaveLinkAdd',
'TicketSlaveLinkDelete',
'TicketMasterLinkDelete',
'EscalationResponseTimeNotifyBefore',
'EscalationUpdateTimeNotifyBefore',
'EscalationSolutionTimeNotifyBefore',
'EscalationResponseTimeStart',
'EscalationUpdateTimeStart',
'EscalationSolutionTimeStart',
'EscalationResponseTimeStop',
'EscalationUpdateTimeStop',
'EscalationSolutionTimeStop',
'NotificationNewTicket',
'NotificationFollowUp',
'NotificationLockTimeout',
'NotificationOwnerUpdate',
'NotificationResponsibleUpdate',
'NotificationAddNote',
'NotificationMove',
'NotificationPendingReminder',
'NotificationEscalation',
'NotificationEscalationNotifyBefore',
'NotificationServiceUpdate'
];
```

Events###Article

List of all article events to be displayed in the GUI.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Events'}->{'Article'} = [
'ArticleCreate',
'ArticleUpdate',
'ArticleSend',
'ArticleBounce',
'ArticleAgentNotification',
'ArticleCustomerNotification',
'ArticleAutoResponse',
'ArticleFlagSet',
'ArticleFlagDelete',
'ArticleAgentNotification',
'ArticleCustomerNotification'
];
```

Events###Queue

List of all queue events to be displayed in the GUI.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Events'}->{'Queue'} = [
'QueueCreate',
'QueueUpdate'
];
```

Ticket → Frontend::Admin::AdminNotificationEvent

Frontend::Admin::AdminNotificationEvent###RichText

Uses richtext for viewing and editing ticket notification.

Standardwert:

```
$Self->{'Frontend::Admin::AdminNotificationEvent'}->{'RichText'} = '1';
```

Frontend::Admin::AdminNotificationEvent###RichTextWidth

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Frontend::Admin::AdminNotificationEvent'}->{'RichTextWidth'} = '620';
```

Frontend::Admin::AdminNotificationEvent###RichTextHeight

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Frontend::Admin::AdminNotificationEvent'}->{'RichTextHeight'} = '320';
```

Notification::Transport###Email

Defines all the parameters for this notification transport.

Standardwert:

```
$Self->{'Notification::Transport'}->{'Email'} = {
  'AgentEnabledByDefault' => '1',
  'Icon' => 'fa fa-envelope',
  'IsOTRSBusinessTransport' => '0',
  'Module' => 'Kernel::System::Ticket::Event::NotificationEvent::Transport::Email',
  'Name' => 'Email',
  'Prio' => '100'
};
```

Notification::Transport###NotificationView

Defines all the parameters for this notification transport.

Standardwert:

```
$Self->{'Notification::Transport'}->{'NotificationView'} = {
  'AgentEnabledByDefault' => '0',
  'Icon' => 'fa fa-th-list',
  'IsOTRSBusinessTransport' => '1',
  'Module' =>
  'Kernel::System::Ticket::Event::NotificationEvent::Transport::NotificationView',
  'Name' => 'Web View',
  'Prio' => '110'
};
```

Notification::Transport###SMS

Defines all the parameters for this notification transport.

Standardwert:

```
$Self->{'Notification::Transport'}->{'SMS'} = {
  'AgentEnabledByDefault' => '0',
  'Icon' => 'fa fa-mobile',
  'IsOTRSBusinessTransport' => '1',
  'Module' => 'Kernel::System::Ticket::Event::NotificationEvent::Transport::SMS',
  'Name' => 'SMS (Short Message Service)',
  'Prio' => '120'
};
```

Notification::CharactersPerLine

Defines the number of character per line used in case an HTML article preview replacement on TemplateGenerator for EventNotifications.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Notification::CharactersPerLine'} = '80';
```

Ticket → Frontend::Admin::ModuleRegistration

Frontend::Module###AdminACL

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminACL'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.ACL.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.ACL.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Configure and manage ACLs.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Access Control Lists (ACL)',
    'Prio' => '750'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Access Control Lists (ACL)'
};
```

Frontend::Module###AdminQueue

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminQueue'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.Queue.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Create and manage queues.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Queues',
    'Prio' => '100'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Queues'
};
```

Frontend::Module###AdminTemplate

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminTemplate'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.Template.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Create and manage templates.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Templates',
    'Prio' => '200'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Templates'
};
```

Frontend::Module###AdminQueueTemplates

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminQueueTemplates'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.QueueTemplates.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Link templates to queues.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Templates ↔ Queues',
    'Prio' => '300'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Templates ↔ Queues'
};
```

Frontend::Module###AdminAutoResponse

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminAutoResponse'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.AutoResponse.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Create and manage responses that are automatically sent.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Auto Responses',
    'Prio' => '400'
  },
};
```

```
'NavBarName' => 'Admin',  
'Title' => 'Auto Responses'  
};
```

Frontend::Module###AdminQueueAutoResponse

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminQueueAutoResponse'} = {  
  'Description' => 'This module is part of the admin area of OTRS.',  
  'Group' => [  
    'admin'  
  ],  
  'Loader' => {  
    'JavaScript' => [  
      'Core.Agent.Admin.QueueAutoResponse.js'  
    ]  
  },  
  'NavBarModule' => {  
    'Block' => 'Queue',  
    'Description' => 'Link queues to auto responses.',  
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',  
    'Name' => 'Auto Responses ↔ Queues',  
    'Prio' => '500'  
  },  
  'NavBarName' => 'Admin',  
  'Title' => 'Auto Responses ↔ Queues'  
};
```

Frontend::Module###AdminAttachment

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminAttachment'} = {  
  'Description' => 'This module is part of the admin area of OTRS.',  
  'Group' => [  
    'admin'  
  ],  
  'Loader' => {  
    'JavaScript' => [  
      'Core.Agent.Admin.Attachment.js'  
    ]  
  },  
  'NavBarModule' => {  
    'Block' => 'Queue',  
    'Description' => 'Create and manage attachments.',  
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',  
    'Name' => 'Attachments',  
    'Prio' => '600'  
  },  
  'NavBarName' => 'Admin',  
  'Title' => 'Attachments'  
};
```

Frontend::Module###AdminTemplateAttachment

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminTemplateAttachment'} = {  
  'Description' => 'This module is part of the admin area of OTRS.',  
  'Group' => [  
    'admin'  
  ],  
  'Loader' => {  
    'JavaScript' => [  
      'Core.Agent.Admin.TemplateAttachment.js'  
    ]  
  }  
};
```

```

},
'NavBarModule' => {
  'Block' => 'Queue',
  'Description' => 'Link attachments to templates.',
  'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
  'Name' => 'Attachments ↔ Templates',
  'Prio' => '700'
},
'NavBarName' => 'Admin',
'Title' => 'Attachments ↔ Templates'
};

```

Frontend::Module###AdminSalutation

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AdminSalutation'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.Salutation.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Create and manage salutations.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Salutations',
    'Prio' => '800'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Salutations'
};

```

Frontend::Module###AdminSignature

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AdminSignature'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.Signature.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Queue',
    'Description' => 'Create and manage signatures.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Signatures',
    'Prio' => '900'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Signatures'
};

```

Frontend::Module###AdminSystemAddress

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AdminSystemAddress'} = {

```



```
'Description' => 'This module is part of the admin area of OTRS.',
'Group' => [
  'admin'
],
'Loader' => {
  'JavaScript' => [
    'Core.Agent.Admin.SystemAddress.js'
  ]
},
'NavBarModule' => {
  'Block' => 'Email',
  'Description' => 'Set sender email addresses for this system.',
  'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
  'Name' => 'Email Addresses',
  'Prio' => '300'
},
'NavBarName' => 'Admin',
'Title' => 'Email Addresses'
};
```

Frontend::Module###AdminNotificationEvent

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminNotificationEvent'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'CSS' => [
      'Core.Agent.Admin.NotificationEvent.css'
    ],
    'JavaScript' => [
      'Core.Agent.Admin.NotificationEvent.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Create and manage ticket notifications.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Ticket Notifications',
    'Prio' => '400'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Ticket Notifications'
};
```

Frontend::Module###AdminService

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminService'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.Service.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Create and manage services.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Services',
    'Prio' => '900'
  },
};
```

```
'NavBarName' => 'Admin',  
'Title' => 'Services'  
};
```

Frontend::Module###AdminSLA

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminSLA'} = {  
  'Description' => 'This module is part of the admin area of OTRS.',  
  'Group' => [  
    'admin'  
  ],  
  'Loader' => {  
    'JavaScript' => [  
      'Core.Agent.Admin.SLA.js'  
    ]  
  },  
  'NavBarModule' => {  
    'Block' => 'Ticket',  
    'Description' => 'Create and manage Service Level Agreements (SLAs).',  
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',  
    'Name' => 'Service Level Agreements',  
    'Prio' => '1000'  
  },  
  'NavBarName' => 'Admin',  
  'Title' => 'Service Level Agreements'  
};
```

Frontend::Module###AdminType

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminType'} = {  
  'Description' => 'This module is part of the admin area of OTRS.',  
  'Group' => [  
    'admin'  
  ],  
  'Loader' => {  
    'JavaScript' => [  
      'Core.Agent.Admin.Type.js'  
    ]  
  },  
  'NavBarModule' => {  
    'Block' => 'Ticket',  
    'Description' => 'Create and manage ticket types.',  
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',  
    'Name' => 'Types',  
    'Prio' => '700'  
  },  
  'NavBarName' => 'Admin',  
  'Title' => 'Types'  
};
```

Frontend::Module###AdminState

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AdminState'} = {  
  'Description' => 'This module is part of the admin area of OTRS.',  
  'Group' => [  
    'admin'  
  ],  
  'Loader' => {  
    'JavaScript' => [  
      'Core.Agent.Admin.State.js'  
    ]  
  }  
};
```

```

},
'NavBarModule' => {
  'Block' => 'Ticket',
  'Description' => 'Create and manage ticket states.',
  'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
  'Name' => 'States',
  'Prio' => '800'
},
'NavBarName' => 'Admin',
'Title' => 'States'
};

```

Frontend::Module###AdminPriority

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AdminPriority'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.Priority.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'Ticket',
    'Description' => 'Create and manage ticket priorities.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'Priorities',
    'Prio' => '850'
  },
  'NavBarName' => 'Admin',
  'Title' => 'Priorities'
};

```

Frontend::Module###AdminGenericAgent

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AdminGenericAgent'} = {
  'Description' => 'This module is part of the admin area of OTRS.',
  'Group' => [
    'admin'
  ],
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.Admin.GenericAgent.js'
    ]
  },
  'NavBarModule' => {
    'Block' => 'System',
    'Description' => 'Manage tasks triggered by event or time based execution.',
    'Module' => 'Kernel::Output::HTML::NavBar::ModuleAdmin',
    'Name' => 'GenericAgent',
    'Prio' => '300'
  },
  'NavBarName' => 'Admin',
  'Title' => 'GenericAgent'
};

```

Ticket → Frontend::Agent

Ticket::Frontend::PendingDiffTime

Time in seconds that gets added to the actual time if setting a pending-state (default: 86400 = 1 day).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::PendingDiffTime'} = '86400';
```

Ticket::Frontend::MaxQueueLevel

Define the max depth of queues.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::MaxQueueLevel'} = '5';
```

Ticket::Frontend::ListType

Shows existing parent/child queue lists in the system in the form of a tree or a list.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ListType'} = 'tree';
```

Ticket::Frontend::TextAreaEmail

Permitted width for compose email windows.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::TextAreaEmail'} = '82';
```

Ticket::Frontend::TextAreaNote

Permitted width for compose note windows.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::TextAreaNote'} = '78';
```

Ticket::Frontend::InformAgentMaxSize

Max size (in rows) of the informed agents box in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::InformAgentMaxSize'} = '3';
```

Ticket::Frontend::InvolvedAgentMaxSize

Max size (in rows) of the involved agents box in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::InvolvedAgentMaxSize'} = '3';
```

Ticket::Frontend::CustomerInfoCompose

Shows the customer user information (phone and email) in the compose screen.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerInfoCompose'} = '1';
```

Ticket::Frontend::CustomerInfoComposeMaxSize

Max size (in characters) of the customer information table (phone and email) in the compose screen.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerInfoComposeMaxSize'} = '22';
```

Ticket::Frontend::CustomerInfoZoom

Shows the customer user's info in the ticket zoom view.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerInfoZoom'} = '1';
```

Ticket::Frontend::CustomerInfoZoomMaxSize

Maximum size (in characters) of the customer information table in the ticket zoom view.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerInfoZoomMaxSize'} = '22';
```

Ticket::Frontend::DynamicFieldsZoomMaxSizeSidebar

Maximum length (in characters) of the dynamic field in the sidebar of the ticket zoom view.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::DynamicFieldsZoomMaxSizeSidebar'} = '18';
```

Ticket::Frontend::DynamicFieldsZoomMaxSizeArticle

Maximum length (in characters) of the dynamic field in the article of the ticket zoom view.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::DynamicFieldsZoomMaxSizeArticle'} = '160';
```

Ticket::Frontend::AccountTime

Activates time accounting.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AccountTime'} = '1';
```

Ticket::Frontend::TimeUnits

Sets the preferred time units (e.g. work units, hours, minutes).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::TimeUnits'} = ' (work units)';
```

Ticket::Frontend::NeedAccountedTime

Defines if time accounting is mandatory in the agent interface. If activated, a note must be entered for all ticket actions (no matter if the note itself is configured as active or is originally mandatory for the individual ticket action screen).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::NeedAccountedTime'} = '0';
```

Ticket::Frontend::BulkAccountedTime

Defines if time accounting must be set to all tickets in bulk action.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::BulkAccountedTime'} = '1';
```

Ticket::Frontend::NeedSpellCheck

Defines if composed messages have to be spell checked in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::NeedSpellCheck'} = '0';
```

Ticket::Frontend::NewOwnerSelection

Shows an owner selection in phone and email tickets in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::NewOwnerSelection'} = '1';
```

Ticket::Frontend::NewResponsibleSelection

Show a responsible selection in phone and email tickets in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::NewResponsibleSelection'} = '1';
```

Ticket::Frontend::NewQueueSelectionType

Defines the recipient target of the phone ticket and the sender of the email ticket ("Queue" shows all queues, "System address" displays all system addresses) in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::NewQueueSelectionType'} = 'Queue';
```

Ticket::Frontend::NewQueueSelectionString

Determines the strings that will be shown as recipient (To:) of the phone ticket and as sender (From:) of the email ticket in the agent interface. For Queue as NewQueueSelectionType "<Queue>" shows the names of the queues and for SystemAddress "<Realname> <<Email>>" shows the name and email of the recipient.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::NewQueueSelectionString'} = '<Queue>';
```

Ticket::Frontend::NewQueueOwnSelection

Determines which options will be valid of the recipient (phone ticket) and the sender (email ticket) in the agent interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::NewQueueOwnSelection'} = {
  '1' => 'First Queue',
  '2' => 'Second Queue'
};
```

Ticket::Frontend::ShowCustomerTickets

Shows customer history tickets in AgentTicketPhone, AgentTicketEmail and AgentTicketCustomer.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ShowCustomerTickets'} = '1';
```

NewTicketInNewWindow::Enabled

If enabled, TicketPhone and TicketEmail will be open in new windows.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'NewTicketInNewWindow::Enabled'} = '0';
```

CustomerDBLink

Defines an external link to the database of the customer (e.g. 'http://your-host/customer.php?CID=[% Data.CustomerID %]' or "").

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerDBLink'} = "[% Env("CGIHandle") %]?
Action=AgentCustomerInformationCenter;CustomerID=[% Data.CustomerID | uri %]";
```

CustomerDBLinkTarget

Defines the target attribute in the link to external customer database. E.g. 'target="cdb"'.
 Diese Einstellung kann nicht deaktiviert werden.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerDBLinkTarget'} = '';
```

CustomerDBLinkClass

Defines the target attribute in the link to external customer database. E.g. 'AsPopup PopupType_TicketAction'.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerDBLinkClass'} = '';
```

Frontend::CommonParam###Action

Defines the default used Frontend-Module if no Action parameter given in the url on the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::CommonParam'}->{'Action'} = 'AgentDashboard';
```

Frontend::CommonParam###QueueID

Default queue ID used by the system in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::CommonParam'}->{'QueueID'} = '0';
```

Frontend::CommonParam###TicketID

Default ticket ID used by the system in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Frontend::CommonParam'}->{'TicketID'} = '';
```

DefaultOverviewColumns

General ticket data shown in the ticket overviews (fall-back). Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note that TicketNumber can not be disabled, because it is necessary.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'DefaultOverviewColumns'} = {  
  'Age' => '2',  
  'Changed' => '1',  
  'CustomerID' => '2',  
  'CustomerName' => '1',  
  'CustomerUserID' => '1',  
  'EscalationResponseTime' => '1',  
  'EscalationSolutionTime' => '1',  
  'EscalationTime' => '1',  
  'EscalationUpdateTime' => '1',  
  'Lock' => '2',  
  'Owner' => '2',  
  'PendingTime' => '1',  
  'Priority' => '1',  
  'Queue' => '2',  
  'Responsible' => '1',  
  'SLA' => '1',  
  'Service' => '1',  
  'State' => '2',  
}
```



```
'TicketNumber' => '2',
'Title' => '2',
'Type' => '1'
};
```

Ticket → Frontend::Agent::Dashboard

DashboardBackend###0100-TicketPendingReminder

Parameters for the dashboard backend of the ticket pending reminder overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.

Standardwert:

```
$Self->{'DashboardBackend'}->{'0100-TicketPendingReminder'} = {
  'Attributes' => 'TicketPendingTimeOlderMinutes=1;StateType=pending
reminder;SortBy=PendingTime;OrderBy=Down;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'Created' => '1',
    'CustomerCompanyName' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All tickets with a reminder set where the reminder date has been
reached',
  'Filter' => 'Locked',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::Dashboard::TicketGeneric',
  'Permission' => 'rw',
  'Time' => 'UntilTime',
  'Title' => 'Reminder Tickets'
};
```

DashboardBackend###0110-TicketEscalation

Parameters for the dashboard backend of the ticket escalation overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTL-Local" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.

Standardwert:

```
$Self->{'DashboardBackend'}->{'0110-TicketEscalation'} = {
  'Attributes' =>
  'TicketEscalationTimeOlderMinutes=1;SortBy=EscalationTime;OrderBy=Down;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'Created' => '1',
    'CustomerCompanyName' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All escalated tickets',
  'Filter' => 'All',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::Dashboard::TicketGeneric',
  'Permission' => 'rw',
  'Time' => 'EscalationTime',
  'Title' => 'Escalated Tickets'
};
```

DashboardBackend###0120-TicketNew

Parameters for the dashboard backend of the new tickets overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.

Standardwert:

```
$Self->{'DashboardBackend'}->{'0120-TicketNew'} = {
  'Attributes' => 'StateType=new;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'Created' => '1',
    'CustomerCompanyName' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
  }
};
```

```
'EscalationUpdateTime' => '1',
'Lock' => '1',
'Owner' => '1',
'PendingTime' => '1',
'Priority' => '1',
'Queue' => '1',
'Responsible' => '1',
'SLA' => '1',
'Service' => '1',
'State' => '1',
'TicketNumber' => '2',
'Title' => '2',
'Type' => '1'
},
'Description' => 'All new tickets, these tickets have not been worked on yet',
'Filter' => 'All',
'Group' => '',
'Limit' => '10',
'Module' => 'Kernel::Output::HTML::Dashboard::TicketGeneric',
'Permission' => 'rw',
'Time' => 'Age',
'Title' => 'New Tickets'
};
```

DashboardBackend###0130-TicketOpen

Parameters for the dashboard backend of the open tickets overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.

Standardwert:

```
$Self->{'DashboardBackend'}->{'0130-TicketOpen'} = {
  'Attributes' => 'StateType=open;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'Created' => '1',
    'CustomerCompanyName' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All open tickets, these tickets have already been worked on, but
  need a response',
  'Filter' => 'All',
  'Group' => '',
  'Limit' => '10',
```

```
'Module' => 'Kernel::Output::HTML::Dashboard::TicketGeneric',
'Permission' => 'rw',
'Time' => 'Age',
'Title' => 'Open Tickets / Need to be answered'
};
```

DashboardBackend###0250-TicketStats

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.

Standardwert:

```
$Self->{'DashboardBackend'}->{'0250-TicketStats'} = {
'Block' => 'ContentSmall',
'CacheTTLLocal' => '30',
'Changed' => '1',
'Closed' => '1',
'Default' => '1',
'Group' => '',
'Module' => 'Kernel::Output::HTML::Dashboard::TicketStatsGeneric',
'Permission' => 'rw',
'Title' => '7 Day Stats'
};
```

DashboardBackend###0260-TicketCalendar

Parameters for the dashboard backend of the upcoming events widget 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.

Standardwert:

```
$Self->{'DashboardBackend'}->{'0260-TicketCalendar'} = {
'Block' => 'ContentSmall',
'CacheTTL' => '2',
'Default' => '1',
'Group' => '',
'Limit' => '6',
'Module' => 'Kernel::Output::HTML::Dashboard::Calendar',
'OwnerOnly' => '',
'Permission' => 'rw',
'Title' => 'Upcoming Events'
};
```

DashboardBackend###0270-TicketQueueOverview

Parameters for the dashboard backend of the queue overview widget of the agent interface. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "QueuePermissionGroup" is not mandatory, queues are only listed if they belong to this permission group if you enable it. "States" is a list of states, the key is the sort order of the state in the widget. "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.

Standardwert:

```
$Self->{'DashboardBackend'}->{'0270-TicketQueueOverview'} = {
'Block' => 'ContentLarge',
'CacheTTLLocal' => '0.5',
'Default' => '1',
'Description' => 'Provides a matrix overview of the tickets per state per queue',
'Group' => '',
'Module' => 'Kernel::Output::HTML::Dashboard::TicketQueueOverview',
'Permission' => 'rw',
```

```
'QueuePermissionGroup' => '',
'Sort' => 'SortBy=Age;OrderBy=Up',
'States' => {
  '1' => 'new',
  '4' => 'open',
  '6' => 'pending reminder'
},
'Title' => 'Ticket Queue Overview'
};
```

DashboardBackend###0280-DashboardEventsTicketCalendar

Parameters for the dashboard backend of the ticket events 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.

Standardwert:

```
$Self->{'DashboardBackend'}->{'0280-DashboardEventsTicketCalendar'} = {
  'Block' => 'ContentLarge',
  'CacheTTL' => '0',
  'Default' => '0',
  'Group' => '',
  'Module' => 'Kernel::Output::HTML::Dashboard::EventsTicketCalendar',
  'Title' => 'Events Ticket Calendar'
};
```

AgentCustomerInformationCenter::Backend###0100-CIC-TicketPendingReminder

Parameters for the dashboard backend of the ticket pending reminder overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.

Standardwert:

```
$Self->{'AgentCustomerInformationCenter::Backend'}->{'0100-CIC-TicketPendingReminder'}
= {
  'Attributes' => 'TicketPendingTimeOlderMinutes=1;StateType=pending
reminder;SortBy=PendingTime;OrderBy=Down;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'Created' => '1',
    'CustomerCompanyName' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
```

```

    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All tickets with a reminder set where the reminder date has been
reached',
  'Filter' => 'Locked',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::Dashboard::TicketGeneric',
  'Permission' => 'ro',
  'Time' => 'UntilTime',
  'Title' => 'Reminder Tickets'
};

```

AgentCustomerInformationCenter::Backend###0110-CIC-TicketEscalation

Parameters for the dashboard backend of the ticket escalation overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTL-Local" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.

Standardwert:

```

$self->{'AgentCustomerInformationCenter::Backend'}->{'0110-CIC-TicketEscalation'} = {
  'Attributes' =>
  'TicketEscalationTimeOlderMinutes=1;SortBy=EscalationTime;OrderBy=Down;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'Created' => '1',
    'CustomerCompanyName' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All escalated tickets',
  'Filter' => 'All',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::Dashboard::TicketGeneric',
  'Permission' => 'ro',
  'Time' => 'EscalationTime',
  'Title' => 'Escalated Tickets'
};

```

AgentCustomerInformationCenter::Backend###0120-CIC-TicketNew

Parameters for the dashboard backend of the new tickets overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the

access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.

Standardwert:

```
$Self->{'AgentCustomerInformationCenter::Backend'}->{'0120-CIC-TicketNew'} = {
  'Attributes' => 'StateType=new;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'Created' => '1',
    'CustomerCompanyName' => '1',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '1',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '1',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Description' => 'All new tickets, these tickets have not been worked on yet',
  'Filter' => 'All',
  'Group' => '',
  'Limit' => '10',
  'Module' => 'Kernel::Output::HTML::Dashboard::TicketGeneric',
  'Permission' => 'ro',
  'Time' => 'Age',
  'Title' => 'New Tickets'
};
```

AgentCustomerInformationCenter::Backend###0130-CIC-TicketOpen

Parameters for the dashboard backend of the open tickets overview of the agent interface. "Limit" is the number of entries shown by default. "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin. Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.

Standardwert:

```
$Self->{'AgentCustomerInformationCenter::Backend'}->{'0130-CIC-TicketOpen'} = {
  'Attributes' => 'StateType=open;',
  'Block' => 'ContentLarge',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'DefaultColumns' => {
    'Age' => '2',
    'Changed' => '1',
    'Created' => '1',
    'CustomerCompanyName' => '1',
```

```

'CustomerID' => '1',
'CustomerName' => '1',
'CustomerUserID' => '1',
'EscalationResponseTime' => '1',
'EscalationSolutionTime' => '1',
'EscalationTime' => '1',
'EscalationUpdateTime' => '1',
'Lock' => '1',
'Owner' => '1',
'PendingTime' => '1',
'Priority' => '1',
'Queue' => '1',
'Responsible' => '1',
'SLA' => '1',
'Service' => '1',
'State' => '1',
'TicketNumber' => '2',
'Title' => '2',
'Type' => '1'
},
'Description' => 'All open tickets, these tickets have already been worked on, but
need a response',
'Filter' => 'All',
'Group' => '',
'Limit' => '10',
'Module' => 'Kernel::Output::HTML::Dashboard::TicketGeneric',
'Permission' => 'ro',
'Time' => 'Age',
'Title' => 'Open Tickets / Need to be answered'
};

```

AgentCustomerInformationCenter::Backend###0500-CIC-CustomerIDStatus

Parameters for the dashboard backend of the customer id status widget of the agent interface . "Group" is used to restrict the access to the plugin (e. g. Group: admin;group1;group2;). "Default" determines if the plugin is enabled by default or if the user needs to enable it manually. "CacheTTLLocal" is the cache time in minutes for the plugin.

Standardwert:

```

$self->{'AgentCustomerInformationCenter::Backend'}->{'0500-CIC-CustomerIDStatus'} = {
  'Attributes' => '',
  'Block' => 'ContentSmall',
  'CacheTTLLocal' => '0.5',
  'Default' => '1',
  'Description' => 'Company Status',
  'Group' => '',
  'Module' => 'Kernel::Output::HTML::Dashboard::CustomerIDStatus',
  'Permission' => 'ro',
  'Title' => 'Company Status'
};

```

Ticket → Frontend::Agent::Dashboard::EventsTicketCalendar

DashboardEventsTicketCalendar###CalendarWidth

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

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```

$self->{'DashboardEventsTicketCalendar'}->{'CalendarWidth'} = '95';

```

DashboardEventsTicketCalendar###Queues

Defines queues that's tickets are used for displaying as calendar events.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:


```
$Self->{'DashboardEventsTicketCalendar'}->{'Queues'} = [
  'Raw'
];
```

DashboardEventsTicketCalendar::DynamicFieldStartTime

Define dynamic field name for start time. This field has to be manually added to the system as Ticket: "Date / Time" and must be activated in ticket creation screens and/or in any other ticket action screens.

Standardwert:

```
$Self->{'DashboardEventsTicketCalendar::DynamicFieldStartTime'} =
  'TicketCalendarStartTime';
```

DashboardEventsTicketCalendar::DynamicFieldEndTime

Define dynamic field name for end time. This field has to be manually added to the system as Ticket: "Date / Time" and must be activated in ticket creation screens and/or in any other ticket action screens.

Standardwert:

```
$Self->{'DashboardEventsTicketCalendar::DynamicFieldEndTime'} =
  'TicketCalendarEndTime';
```

DashboardEventsTicketCalendar::DynamicFieldsForEvents

Defines the dynamic fields that are used for displaying on calendar events.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'DashboardEventsTicketCalendar::DynamicFieldsForEvents'} = [
  'TicketCalendarStartTime',
  'TicketCalendarEndTime'
];
```

DashboardEventsTicketCalendar::TicketFieldsForEvents

Defines the ticket fields that are going to be displayed calendar events. The "Key" defines the field or ticket attribute and the "Content" defines the display name.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'DashboardEventsTicketCalendar::TicketFieldsForEvents'} = {
  'CustomerID' => 'Customer ID',
  'CustomerUserID' => 'Customer user',
  'Priority' => 'Priority',
  'Queue' => 'Queue',
  'SLA' => 'SLA',
  'Service' => 'Service',
  'State' => 'State',
  'Title' => 'Title',
  'Type' => 'Type'
};
```

Ticket → Frontend::Agent::Dashboard::TicketFilters

OnlyValuesOnTicket

Defines if the values for filters should be retrieved from all available tickets. If set to "Yes", only values which are actually used in any ticket will be available for filtering. Please note: The list of customers will always be retrieved like this.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'OnlyValuesOnTicket'} = '1';
```

Ticket → Frontend::Agent::LinkObject

LinkObject::ComplexTable::SettingsVisibility###Ticket

Define Actions where a settings button is available in the linked objects widget (LinkObject::ViewMode = "complex"). Please note that these Actions must have registered the following JS and CSS files: Core.AllocationList.css, Core.UI.AllocationList.js, Core.UI.Table.Sort.js, Core.Agent.TableFilters.js.

Standardwert:

```
$Self->{'LinkObject::ComplexTable::SettingsVisibility'}->{'Ticket'} = [
  'AgentTicketZoom'
];
```

LinkObject::ComplexTable###Ticket

Define which columns are shown in the linked tickets widget (LinkObject::ViewMode = "complex"). Note: Only Ticket attributes and Dynamic Fields (DynamicField_NameX) are allowed for DefaultColumns. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.

Standardwert:

```
$Self->{'LinkObject::ComplexTable'}->{'Ticket'} = {
  'DefaultColumns' => {
    'Age' => '1',
    'Changed' => '1',
    'Created' => '2',
    'CustomerID' => '1',
    'CustomerName' => '1',
    'CustomerUserID' => '1',
    'EscalationResponseTime' => '1',
    'EscalationSolutionTime' => '1',
    'EscalationTime' => '1',
    'EscalationUpdateTime' => '1',
    'Lock' => '1',
    'Owner' => '1',
    'PendingTime' => '1',
    'Priority' => '1',
    'Queue' => '2',
    'Responsible' => '1',
    'SLA' => '1',
    'Service' => '1',
    'State' => '2',
    'TicketNumber' => '2',
    'Title' => '2',
    'Type' => '1'
  },
  'Module' => 'Kernel::Output::HTML::LinkObject::Ticket.pm',
  'Priority' => {
    'Age' => '110',
    'Changed' => '120',
    'Created' => '310',
    'CustomerID' => '240',
    'CustomerName' => '250',
    'CustomerUserID' => '260',
    'EscalationResponseTime' => '160',
    'EscalationSolutionTime' => '150',
    'EscalationTime' => '140',
    'EscalationUpdateTime' => '170',
    'Lock' => '200',
    'Owner' => '220',
    'PendingTime' => '130',
    'Priority' => '300',
    'Queue' => '210',
    'Responsible' => '230',
    'SLA' => '290',
    'Service' => '280',
```

```
'State' => '190',
'TicketNumber' => '100',
'Title' => '180',
'Type' => '270'
}
};
```

Ticket → Frontend::Agent::ModuleMetaHead

Frontend::HeaderMetaModule###2-TicketSearch

Module to generate html OpenSearch profile for short ticket search in the agent interface.

Standardwert:

```
$Self->{'Frontend::HeaderMetaModule'}->{'2-TicketSearch'} = {
  'Action' => 'AgentTicketSearch',
  'Module' => 'Kernel::Output::HTML::HeaderMeta::AgentTicketSearch'
};
```

Ticket → Frontend::Agent::ModuleNotify

Frontend::NotifyModule###5000-Ticket::TicketEscalation

Module to show notifications and escalations (ShownMax: max. shown escalations, EscalationInMinutes: Show ticket which will escalation in, CacheTime: Cache of calculated escalations in seconds).

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::NotifyModule'}->{'5000-Ticket::TicketEscalation'} = {
  'CacheTime' => '40',
  'EscalationInMinutes' => '120',
  'Module' => 'Kernel::Output::HTML::Notification::AgentTicketEscalation',
  'ShownMax' => '25'
};
```

Ticket → Frontend::Agent::ModuleRegistration

Frontend::Module###AgentTicketQueue

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentTicketQueue'} = {
  'Description' => 'Overview of all open Tickets.',
  'Loader' => {
    'CSS' => [
      'Core.AgentTicketQueue.css',
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.UI.AllocationList.js',
      'Core.Agent.TableFilters.js',
      'Core.Agent.Overview.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'o',
      'Block' => '',
      'Description' => 'Overview of all open Tickets.',
      'Link' => 'Action=AgentTicketQueue',
      'LinkOption' => '',
      'Name' => 'Queue view',
      'NavBar' => 'Ticket',
      'Prio' => '100',
    }
  ]
};
```

```

    'Type' => ''
  },
  {
    'AccessKey' => 't',
    'Block' => 'ItemArea',
    'Description' => '',
    'Link' => 'Action=AgentTicketQueue',
    'LinkOption' => '',
    'Name' => 'Tickets',
    'NavBar' => 'Ticket',
    'Prio' => '200',
    'Type' => 'Menu'
  }
],
'NavBarName' => 'Ticket',
'Title' => 'QueueView'
};

```

Frontend::Module###AgentTicketService

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketService'} = {
  'Description' => 'Overview of all open Tickets.',
  'Loader' => {
    'CSS' => [
      'Core.AgentTicketService.css',
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.UI.AllocationList.js',
      'Core.Agent.TableFilters.js',
      'Core.Agent.Overview.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => '0',
      'Block' => '',
      'Description' => 'Overview of all open Tickets.',
      'Link' => 'Action=AgentTicketService',
      'LinkOption' => '',
      'Name' => 'Service view',
      'NavBar' => 'Ticket',
      'Prio' => '105',
      'Type' => ''
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'ServiceView'
};

```

Frontend::Module###AgentTicketPhone

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketPhone'} = {
  'Description' => 'Create new phone ticket.',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.CustomerSearch.js',
      'Core.Agent.CustomerSearchAutoComplete.js',
      'Core.Agent.TicketAction.js',
      'Core.Agent.TicketPhone.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'
};

```

Frontend::Module###AgentTicketPhoneOutbound

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketPhoneOutbound'} = {
  'Description' => 'Phone Call.',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketAction.js',
      'Core.Agent.TicketPhoneCommon.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Phone-Ticket'
};

```

Frontend::Module###AgentTicketPhoneInbound

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketPhoneInbound'} = {
  'Description' => 'Incoming Phone Call.',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketAction.js',
      'Core.Agent.TicketPhoneCommon.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Phone-Ticket'
};

```

Frontend::Module###AgentTicketEmail

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketEmail'} = {
  'Description' => 'Create new email ticket.',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.CustomerSearch.js',
      'Core.Agent.CustomerSearchAutoComplete.js',
      'Core.Agent.TicketAction.js',
      'Core.Agent.TicketEmail.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'm',
      'Block' => '',
      'Description' => 'Create new email ticket and send this out (outbound).',
      'Link' => 'Action=AgentTicketEmail',
      'LinkOption' => '',
      'Name' => 'New email ticket',

```

```

    'NavBar' => 'Ticket',
    'Prio' => '210',
    'Type' => ''
  }
],
'NavBarName' => 'Ticket',
'Title' => 'New email ticket'
};

```

Frontend::Module###AgentTicketSearch

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketSearch'} = {
  'Description' => 'Search Ticket.',
  'Loader' => {
    'JavaScript' => [
      'Core.UI.AllocationList.js',
      'Core.Agent.TableFilters.js',
      'Core.Agent.Overview.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 's',
      'Block' => '',
      'Description' => 'Search Tickets.',
      'Link' => 'Action=AgentTicketSearch',
      'LinkOption' => 'onclick="window.setTimeout(function()
{Core.Agent.Search.OpenSearchDialog(\'AgentTicketSearch\');}, 0); return false;"',
      'Name' => 'Search',
      'NavBar' => 'Ticket',
      'Prio' => '300',
      'Type' => ''
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'Search'
};

```

Frontend::Module###AgentTicketLockedView

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketLockedView'} = {
  'Description' => 'Locked Tickets.',
  'Loader' => {
    'CSS' => [
      'Core.AgentTicketQueue.css',
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.UI.AllocationList.js',
      'Core.Agent.TableFilters.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Locked Tickets'
};

```

Frontend::Module###AgentTicketResponsibleView

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketResponsibleView'} = {
  'Description' => 'Responsible Tickets.',
  'Loader' => {
    'CSS' => [

```

```

    'Core.AllocationList.css'
  ],
  'JavaScript' => [
    'Core.UI.AllocationList.js',
    'Core.Agent.TableFilters.js',
    'Core.Agent.Overview.js'
  ]
},
'NavBarName' => 'Ticket',
'Title' => 'Responsible Tickets'
};

```

Frontend::Module###AgentTicketWatchView

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketWatchView'} = {
  'Description' => 'Watched Tickets.',
  'Loader' => {
    'CSS' => [
      'Core.AgentTicketQueue.css',
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.UI.AllocationList.js',
      'Core.Agent.TableFilters.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Watched Tickets'
};

```

Frontend::Module###AgentCustomerSearch

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentCustomerSearch'} = {
  'Description' => 'Agent Customer Search.',
  'NavBarName' => 'Ticket',
  'Title' => 'Agent Customer Search'
};

```

Frontend::Module###AgentUserSearch

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentUserSearch'} = {
  'Description' => 'Agent User Search.',
  'NavBarName' => 'Ticket',
  'Title' => 'Agent User Search'
};

```

Frontend::Module###AgentTicketStatusView

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketStatusView'} = {
  'Description' => 'Overview of all open tickets.',
  'Loader' => {
    'CSS' => [
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.UI.AllocationList.js',
      'Core.Agent.TableFilters.js',
      'Core.Agent.Overview.js'
    ]
  }
};

```

```

    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'v',
      'Block' => '',
      'Description' => 'Overview of all open Tickets.',
      'Link' => 'Action=AgentTicketStatusView',
      'LinkOption' => '',
      'Name' => 'Status view',
      'NavBar' => 'Ticket',
      'Prio' => '110',
      'Type' => ''
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'Status view'
};

```

Frontend::Module###AgentTicketEscalationView

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketEscalationView'} = {
  'Description' => 'Overview of all escalated tickets.',
  'Loader' => {
    'CSS' => [
      'Core.AllocationList.css'
    ],
    'JavaScript' => [
      'Core.UI.AllocationList.js',
      'Core.Agent.TableFilters.js',
      'Core.Agent.Overview.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'e',
      'Block' => '',
      'Description' => 'Overview Escalated Tickets.',
      'Link' => 'Action=AgentTicketEscalationView',
      'LinkOption' => '',
      'Name' => 'Escalation view',
      'NavBar' => 'Ticket',
      'Prio' => '120',
      'Type' => ''
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'Escalation view'
};

```

Frontend::Module###AgentZoom

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentZoom'} = {
  'Description' => 'Compat module for AgentZoom to AgentTicketZoom.',
  'NavBarName' => 'Ticket',
  'Title' => ''
};

```

Frontend::Module###AgentTicketZoom

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketZoom'} = {
  'Description' => 'Ticket Zoom.',

```



```
'Loader' => {
  'CSS' => [
    'Core.Agent.TicketProcess.css',
    'Core.Agent.TicketMenuModuleCluster.css',
    'Core.AllocationList.css'
  ],
  'JavaScript' => [
    'thirdparty/jquery-tablesorter-2.0.5/jquery.tablesorter.js',
    'Core.Agent.TicketZoom.js',
    'Core.UI.AllocationList.js',
    'Core.UI.Table.Sort.js',
    'Core.Agent.TableFilters.js',
    'Core.Agent.LinkObject.js'
  ]
},
'NavBarName' => 'Ticket',
'Title' => 'Zoom'
};
```

Frontend::Module###AgentTicketAttachment

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentTicketAttachment'} = {
  'Description' => 'To download attachments.',
  'NavBarName' => 'Ticket',
  'Title' => ''
};
```

Frontend::Module###AgentTicketPlain

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentTicketPlain'} = {
  'Description' => 'Ticket plain view of an email.',
  'NavBarName' => 'Ticket',
  'Title' => 'Plain'
};
```

Frontend::Module###AgentTicketNote

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentTicketNote'} = {
  'Description' => 'Ticket Note.',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketAction.js',
      'Core.Agent.TicketActionCommon.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Note'
};
```

Frontend::Module###AgentTicketMerge

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentTicketMerge'} = {
  'Description' => 'Ticket Merge.',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketMerge.js'
    ]
  },
};
```

```
'NavBarName' => 'Ticket',  
'Title' => 'Merge'  
};
```

Frontend::Module###AgentTicketPending

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentTicketPending'} = {  
  'Description' => 'Ticket Pending.',  
  'Loader' => {  
    'JavaScript' => [  
      'Core.Agent.TicketAction.js',  
      'Core.Agent.TicketActionCommon.js'  
    ]  
  },  
  'NavBarName' => 'Ticket',  
  'Title' => 'Pending'  
};
```

Frontend::Module###AgentTicketWatcher

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentTicketWatcher'} = {  
  'Description' => 'A TicketWatcher Module.',  
  'NavBarName' => 'Ticket-Watcher',  
  'Title' => 'Ticket Watcher'  
};
```

Frontend::Module###AgentTicketPriority

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentTicketPriority'} = {  
  'Description' => 'Ticket Priority.',  
  'Loader' => {  
    'JavaScript' => [  
      'Core.Agent.TicketAction.js',  
      'Core.Agent.TicketActionCommon.js'  
    ]  
  },  
  'NavBarName' => 'Ticket',  
  'Title' => 'Priority'  
};
```

Frontend::Module###AgentTicketLock

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentTicketLock'} = {  
  'Description' => 'Ticket Lock.',  
  'NavBarName' => 'Ticket',  
  'Title' => 'Lock'  
};
```

Frontend::Module###AgentTicketMove

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentTicketMove'} = {  
  'Description' => 'Ticket Move.',  
  'Loader' => {  
    'JavaScript' => [  

```

```

    'Core.Agent.TicketAction.js',
    'Core.Agent.TicketMove.js'
  ]
},
'NavBarName' => 'Ticket',
'Title' => 'Move'
};

```

Frontend::Module###AgentTicketHistory

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketHistory'} = {
  'Description' => 'Ticket History.',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketHistory.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'History'
};

```

Frontend::Module###AgentTicketOwner

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketOwner'} = {
  'Description' => 'Ticket Owner.',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketAction.js',
      'Core.Agent.TicketActionCommon.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Owner'
};

```

Frontend::Module###AgentTicketResponsible

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketResponsible'} = {
  'Description' => 'Ticket Responsible.',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketAction.js',
      'Core.Agent.TicketActionCommon.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Responsible'
};

```

Frontend::Module###AgentTicketCompose

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketCompose'} = {
  'Description' => 'Ticket Compose email Answer.',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.CustomerSearch.js',
      'Core.Agent.CustomerSearchAutoComplete.js',
      'Core.Agent.TicketAction.js',

```

```

    'Core.Agent.TicketCompose.js'
  ]
},
'NavBarName' => 'Ticket',
'Title' => 'Compose'
};

```

Frontend::Module###AgentTicketBounce

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketBounce'} = {
  'Description' => 'Ticket Compose Bounce Email.',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.TicketBounce.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Bounce'
};

```

Frontend::Module###AgentTicketForward

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketForward'} = {
  'Description' => 'Ticket Forward Email.',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.CustomerSearch.js',
      'Core.Agent.CustomerSearchAutoComplete.js',
      'Core.Agent.TicketAction.js',
      'Core.Agent.TicketForward.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Forward'
};

```

Frontend::Module###AgentTicketEmailOutbound

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketEmailOutbound'} = {
  'Description' => 'Ticket Outbound Email.',
  'Loader' => {
    'JavaScript' => [
      'Core.Agent.CustomerSearch.js',
      'Core.Agent.CustomerSearchAutoComplete.js',
      'Core.Agent.TicketAction.js',
      'Core.Agent.TicketEmailOutbound.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Email Outbound'
};

```

Frontend::Module###AgentTicketCustomer

Frontend module registration for the agent interface.

Standardwert:

```

$self->{'Frontend::Module'}->{'AgentTicketCustomer'} = {
  'Description' => 'Ticket Customer.',
  'Loader' => {
    'JavaScript' => [

```

```
'Core.Agent.CustomerSearch.js',  
'Core.Agent.CustomerSearchAutoComplete.js',  
'Core.Agent.TicketAction.js'  
]  
},  
'NavBarName' => 'Ticket',  
'Title' => 'Customer'  
};
```

Frontend::Module###AgentTicketClose

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentTicketClose'} = {  
  'Description' => 'Ticket Close.',  
  'Loader' => {  
    'JavaScript' => [  
      'Core.Agent.TicketAction.js',  
      'Core.Agent.TicketActionCommon.js'  
    ]  
  },  
  'NavBarName' => 'Ticket',  
  'Title' => 'Close'  
};
```

Frontend::Module###AgentTicketFreeText

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentTicketFreeText'} = {  
  'Description' => 'Ticket FreeText.',  
  'Loader' => {  
    'JavaScript' => [  
      'Core.Agent.TicketAction.js',  
      'Core.Agent.TicketActionCommon.js'  
    ]  
  },  
  'NavBarName' => 'Ticket',  
  'Title' => 'Free Fields'  
};
```

Frontend::Module###AgentTicketPrint

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentTicketPrint'} = {  
  'Description' => 'Ticket Print.',  
  'NavBarName' => 'Ticket',  
  'Title' => 'Print'  
};
```

Frontend::Module###AgentTicketBulk

Frontend module registration for the agent interface.

Standardwert:

```
$Self->{'Frontend::Module'}->{'AgentTicketBulk'} = {  
  'Description' => 'Ticket bulk module.',  
  'Loader' => {  
    'JavaScript' => [  
      'Core.Agent.TicketAction.js',  
      'Core.Agent.TicketBulk.js'  
    ]  
  },  
  'NavBarName' => 'Ticket',  
  'Title' => 'Bulk Action'  
};
```

Ticket → Frontend::Agent::Preferences

PreferencesGroups###CustomQueue

Parameters for the CustomQueue object in the preference view of the agent interface.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'CustomQueue'} = {
  'Active' => '1',
  'Column' => 'Notification Settings',
  'Desc' => 'Your queue selection of your preferred queues. You also get notified about those queues via email if enabled.',
  'Key' => '',
  'Label' => 'My Queues',
  'Module' => 'Kernel::Output::HTML::Preferences::CustomQueue',
  'Permission' => 'ro',
  'Prio' => '1000'
};
```

PreferencesGroups###CustomService

Parameters for the CustomService object in the preference view of the agent interface.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'CustomService'} = {
  'Active' => '1',
  'Column' => 'Notification Settings',
  'Desc' => 'Your service selection of your preferred services. You also get notified about those services via email if enabled.',
  'Key' => '',
  'Label' => 'My Services',
  'Module' => 'Kernel::Output::HTML::Preferences::CustomService',
  'Prio' => '1000'
};
```

PreferencesGroups###RefreshTime

Parameters for the RefreshTime object in the preference view of the agent interface.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'RefreshTime'} = {
  'Active' => '1',
  'Column' => 'Other Settings',
  'Data' => {
    '0' => 'off',
    '10' => '10 minutes',
    '15' => '15 minutes',
    '2' => ' 2 minutes',
    '5' => ' 5 minutes',
    '7' => ' 7 minutes'
  },
  'DataSelected' => '0',
  'Desc' => 'If enabled, the different overviews (Dashboard, LockedView, QueueView) will automatically refresh after the specified time.',
  'Key' => 'After',
  'Label' => 'Overview Refresh Time',
  'Module' => 'Kernel::Output::HTML::Preferences::Generic',
  'PrefKey' => 'UserRefreshTime',
  'Prio' => '2000'
};
```

PreferencesGroups###TicketOverviewSmallPageShown

Parameters for the pages (in which the tickets are shown) of the small ticket overview.

Standardwert:

```
$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::Preferences::Generic',
'PrefKey' => 'UserTicketOverviewSmallPageShown',
'Prio' => '8000'
};
```

PreferencesGroups###TicketOverviewFilterSettings

Parameters for the column filters of the small ticket overview.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'TicketOverviewFilterSettings'} = {
'Active' => '0',
'Column' => 'Other Settings',
'Key' => 'Column ticket filters for Ticket Overviews type "Small".',
'Label' => 'Enabled filters.',
'Module' => 'Kernel::Output::HTML::Preferences::ColumnFilters',
'PrefKey' => 'UserFilterColumnsEnabled',
'Prio' => '8100'
};
```

PreferencesGroups###TicketOverviewMediumPageShown

Parameters for the pages (in which the tickets are shown) of the medium ticket overview.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'TicketOverviewMediumPageShown'} = {
'Active' => '0',
'Column' => 'Other Settings',
'Data' => {
'10' => '10',
'15' => '15',
'20' => '20',
'25' => '25',
'30' => '30',
'35' => '35'
},
'DataSelected' => '20',
'Key' => 'Ticket limit per page for Ticket Overview "Medium"',
'Label' => 'Ticket Overview "Medium" Limit',
'Module' => 'Kernel::Output::HTML::Preferences::Generic',
'PrefKey' => 'UserTicketOverviewMediumPageShown',
'Prio' => '8100'
};
```

PreferencesGroups###TicketOverviewPreviewPageShown

Parameters for the pages (in which the tickets are shown) of the ticket preview overview.

Standardwert:

```
$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::Preferences::Generic',
'PrefKey' => 'UserTicketOverviewPreviewPageShown',
'Prio' => '8200'
};
```

PreferencesGroups###CreateNextMask

Parameters for the CreateNextMask object in the preference view of the agent interface.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'CreateNextMask'} = {
  'Active' => '1',
  'Column' => 'Other Settings',
  'Data' => {
    '0' => 'CreateTicket',
    'AgentTicketZoom' => 'TicketZoom'
  },
  'DataSelected' => '',
  'Desc' => 'Configure which screen should be shown after a new ticket has been created.',
  'Key' => 'Screen',
  'Label' => 'Screen after new ticket',
  'Module' => 'Kernel::Output::HTML::Preferences::Generic',
  'PrefKey' => 'UserCreateNextMask',
  'Prio' => '3000'
};
```

PreferencesGroups###NotificationEvent

Transport selection for ticket notifications.

Standardwert:

```
$Self->{'PreferencesGroups'}->{'NotificationEvent'} = {
  'Active' => '1',
  'Column' => 'Notification Settings',
  'Desc' => 'Choose for which kind of ticket changes you want to receive notifications.',
  'Label' => 'Ticket notifications',
  'Module' => 'Kernel::Output::HTML::Preferences::NotificationEvent',
  'PrefKey' => 'AdminNotificationEventTransport',
  'Prio' => '8000'
};
```

Ticket → Frontend::Agent::SearchRouter

Frontend::Search###AgentCustomerInformationCenter

Search backend router.

Standardwert:

```
$Self->{'Frontend::Search'}->{'AgentCustomerInformationCenter'} = {
  '^AgentCustomerInformationCenter' => 'Action=AgentCustomerInformationCenterSearch'
};
```

Frontend::Search::JavaScript###AgentCustomerInformationCenter

JavaScript function for the search frontend.

Standardwert:

```
$Self->{'Frontend::Search::JavaScript'}->{'AgentCustomerInformationCenter'} = {
  '^AgentCustomerInformationCenter' =>
  'Core.Agent.CustomerInformationCenterSearch.OpenSearchDialog()'
};
```


Frontend::Search###Ticket

Search backend router.

Standardwert:

```
$Self->{'Frontend::Search'}->{'Ticket'} = {  
  '^AgentTicket' => 'Action=AgentTicketSearch;Subaction=AJAX'  
};
```

Ticket → Frontend::Agent::Ticket::ArticleAttachmentModule

Ticket::Frontend::ArticleAttachmentModule###1-Download

Shows a link to download article attachments in the zoom view of the article in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ArticleAttachmentModule'}->{'1-Download'} = {  
  'Module' => 'Kernel::Output::HTML::ArticleAttachment::Download'  
};
```

Ticket::Frontend::ArticleAttachmentModule###2-HTML-Viewer

Shows a link to access article attachments via a html online viewer in the zoom view of the article in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ArticleAttachmentModule'}->{'2-HTML-Viewer'} = {  
  'Module' => 'Kernel::Output::HTML::ArticleAttachment::HTMLViewer'  
};
```

Ticket → Frontend::Agent::Ticket::ArticleComposeModule

Ticket::Frontend::ArticleComposeModule###1-EmailSecurity

Module to define the email security options to use (PGP or S/MIME).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ArticleComposeModule'}->{'1-EmailSecurity'} = {  
  'Module' => 'Kernel::Output::HTML::ArticleCompose::Security',  
  'ParamType' => 'Single'  
};
```

Ticket::Frontend::ArticleComposeModule###2-SignEmail

Module to compose signed messages (PGP or S/MIME).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ArticleComposeModule'}->{'2-SignEmail'} = {  
  'Module' => 'Kernel::Output::HTML::ArticleCompose::Sign',  
  'ParamType' => 'Single'  
};
```

Ticket::Frontend::ArticleComposeModule###3-CryptEmail

Module to encrypt composed messages (PGP or S/MIME).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ArticleComposeModule'}->{'3-CryptEmail'} = {
  'Module' => 'Kernel::Output::HTML::ArticleCompose::Crypt',
  'ParamType' => 'Multiple'
};
```

Ticket → Frontend::Agent::Ticket::ArticleViewModule

Ticket::Frontend::ArticleViewModule###1-PGP

Agent interface article notification module to check PGP.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ArticleViewModule'}->{'1-PGP'} = {
  'Module' => 'Kernel::Output::HTML::ArticleCheck::PGP'
};
```

Ticket::Frontend::ArticleViewModule###1-SMIME

Agent interface module to check incoming emails in the Ticket-Zoom-View if the S/MIME-key is available and true.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ArticleViewModule'}->{'1-SMIME'} = {
  'Module' => 'Kernel::Output::HTML::ArticleCheck::SMIME'
};
```

Ticket → Frontend::Agent::Ticket::ArticleViewModulePre

Ticket::Frontend::ArticlePreViewModule###1-PGP

Agent interface article notification module to check PGP.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ArticlePreViewModule'}->{'1-PGP'} = {
  'Module' => 'Kernel::Output::HTML::ArticleCheck::PGP'
};
```

Ticket::Frontend::ArticlePreViewModule###2-SMIME

Agent interface article notification module to check S/MIME.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ArticlePreViewModule'}->{'2-SMIME'} = {
  'Module' => 'Kernel::Output::HTML::ArticleCheck::SMIME'
};
```

Ticket → Frontend::Agent::Ticket::MenuModule

Ticket::Frontend::MenuModule###000-Back

Shows a link in the menu to go back 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'000-Back'} = {
  'Action' => '',
  'ClusterName' => '',
  'ClusterPriority' => '',
  'Description' => 'Go back',
  'Link' => "[% Env("LastScreenOverview") %];TicketID=[% Data.TicketID | html %]",
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'Back',
  'PopupType' => '',
  'Target' => ''
};
```

Ticket::Frontend::MenuModule###100-Lock

Shows a link in the menu to lock/unlock tickets 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'100-Lock'} = {
  'Action' => 'AgentTicketLock',
  'ClusterName' => 'Miscellaneous',
  'ClusterPriority' => '800',
  'Description' => 'Lock / unlock this ticket',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Lock',
  'Name' => 'Lock',
  'Target' => ''
};
```

Ticket::Frontend::MenuModule###200-History

Shows a link in the menu to access the history of 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'200-History'} = {
  'Action' => 'AgentTicketHistory',
  'ClusterName' => 'Miscellaneous',
  'ClusterPriority' => '800',
  'Description' => 'Show the history for this ticket',
  'Link' => 'Action=AgentTicketHistory;TicketID=[% Data.TicketID | html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'History',
  'PopupType' => 'TicketHistory',
  'Target' => ''
};
```

Ticket::Frontend::MenuModule###210-Print

Shows a link in the menu to print a ticket or an article 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'210-Print'} = {
  'Action' => 'AgentTicketPrint',
  'ClusterName' => '',
  'ClusterPriority' => '',
  'Description' => 'Print this ticket',
};
```

```
'Link' => 'Action=AgentTicketPrint;TicketID=[% Data.TicketID | html %]',
'LinkParam' => 'target="print"',
'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
'Name' => 'Print',
'PopupType' => 'TicketAction',
'Target' => ''
};
```

Ticket::Frontend::MenuModule###300-Priority

Shows a link in the menu to see the priority of 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'300-Priority'} = {
  'Action' => 'AgentTicketPriority',
  'ClusterName' => '',
  'ClusterPriority' => '',
  'Description' => 'Change the priority for this ticket',
  'Link' => 'Action=AgentTicketPriority;TicketID=[% Data.TicketID | html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'Priority',
  'PopupType' => 'TicketAction',
  'Target' => ''
};
```

Ticket::Frontend::MenuModule###310-FreeText

Shows a link in the menu to add a free text field 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'310-FreeText'} = {
  'Action' => 'AgentTicketFreeText',
  'ClusterName' => 'Miscellaneous',
  'ClusterPriority' => '800',
  'Description' => 'Change the free fields for this ticket',
  'Link' => 'Action=AgentTicketFreeText;TicketID=[% Data.TicketID | html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'Free Fields',
  'PopupType' => 'TicketAction',
  'Target' => ''
};
```

Ticket::Frontend::MenuModule###320-Link

Shows a link in the menu that allows linking a ticket with another object 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'320-Link'} = {
  'Action' => 'AgentLinkObject',
  'ClusterName' => 'Miscellaneous',
  'ClusterPriority' => '800',
  'Description' => 'Link this ticket to other objects',
  'Link' => 'Action=AgentLinkObject;SourceObject=Ticket;SourceKey=[% Data.TicketID |
html %]',
};
```

```
'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
'Name' => 'Link',
'PopupType' => 'TicketAction',
'Target' => ''
};
```

Ticket::Frontend::MenuModule###400-Owner

Shows a link in the menu to see the owner of 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'400-Owner'} = {
'Action' => 'AgentTicketOwner',
'ClusterName' => 'People',
'ClusterPriority' => '430',
'Description' => 'Change the owner for this ticket',
'Link' => 'Action=AgentTicketOwner;TicketID=[% Data.TicketID | html %]',
'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
'Name' => 'Owner',
'PopupType' => 'TicketAction',
'Target' => ''
};
```

Ticket::Frontend::MenuModule###410-Responsible

Shows a link in the menu to see the responsible agent of 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'410-Responsible'} = {
'Action' => 'AgentTicketResponsible',
'ClusterName' => 'People',
'ClusterPriority' => '430',
'Description' => 'Change the responsible for this ticket',
'Link' => 'Action=AgentTicketResponsible;TicketID=[% Data.TicketID | html %]',
'Module' => 'Kernel::Output::HTML::TicketMenu::Responsible',
'Name' => 'Responsible',
'PopupType' => 'TicketAction',
'Target' => ''
};
```

Ticket::Frontend::MenuModule###420-Customer

Shows a link in the menu to see the customer who requested the 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'420-Customer'} = {
'Action' => 'AgentTicketCustomer',
'ClusterName' => 'People',
'ClusterPriority' => '430',
'Description' => 'Change the customer for this ticket',
'Link' => 'Action=AgentTicketCustomer;TicketID=[% Data.TicketID | html %]',
'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
```

```
'Name' => 'Customer',
'PopupType' => 'TicketAction',
'Target' => ''
};
```

Ticket::Frontend::MenuModule###420-Note

Shows a link in the menu to add a note 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'420-Note'} = {
  'Action' => 'AgentTicketNote',
  'ClusterName' => 'Communication',
  'ClusterPriority' => '435',
  'Description' => 'Add a note to this ticket',
  'Link' => 'Action=AgentTicketNote;TicketID=[% Data.TicketID | html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'Note',
  'PopupType' => 'TicketAction',
  'Target' => ''
};
```

Ticket::Frontend::MenuModule###425-Phone Call Outbound

Shows a link in the menu to add a note 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'425-Phone Call Outbound'} = {
  'Action' => 'AgentTicketPhoneOutbound',
  'ClusterName' => 'Communication',
  'ClusterPriority' => '435',
  'Description' => 'Add an outbound phone call to this ticket',
  'Link' => 'Action=AgentTicketPhoneOutbound;TicketID=[% Data.TicketID | html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'Phone Call Outbound',
  'PopupType' => 'TicketAction',
  'Target' => ''
};
```

Ticket::Frontend::MenuModule###426-Phone Call Inbound

Shows a link in the menu to add a note 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'426-Phone Call Inbound'} = {
  'Action' => 'AgentTicketPhoneInbound',
  'ClusterName' => 'Communication',
  'ClusterPriority' => '435',
  'Description' => 'Add an inbound phone call to this ticket',
  'Link' => 'Action=AgentTicketPhoneInbound;TicketID=[% Data.TicketID | html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'Phone Call Inbound',
  'PopupType' => 'TicketAction',
  'Target' => ''
};
```

Ticket::Frontend::MenuModule###427-Email Outbound

Shows a link in the menu to send an outbound email 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'427-Email Outbound'} = {
  'Action' => 'AgentTicketEmailOutbound',
  'ClusterName' => 'Communication',
  'ClusterPriority' => '435',
  'Description' => 'Send new outgoing mail from this ticket',
  'Link' => 'Action=AgentTicketEmailOutbound;TicketID=[% Data.TicketID | html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'E-Mail Outbound',
  'PopupType' => 'TicketAction',
  'Target' => ''
};
```

Ticket::Frontend::MenuModule###430-Merge

Shows a link in the menu that allows merging tickets 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'430-Merge'} = {
  'Action' => 'AgentTicketMerge',
  'ClusterName' => 'Miscellaneous',
  'ClusterPriority' => '800',
  'Description' => 'Merge this ticket and all articles into a another ticket',
  'Link' => 'Action=AgentTicketMerge;TicketID=[% Data.TicketID | html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'Merge',
  'PopupType' => 'TicketAction',
  'Target' => ''
};
```

Ticket::Frontend::MenuModule###440-Pending

Shows a link in the menu to set a ticket as pending 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'440-Pending'} = {
  'Action' => 'AgentTicketPending',
  'ClusterName' => '',
  'ClusterPriority' => '',
  'Description' => 'Set this ticket to pending',
  'Link' => 'Action=AgentTicketPending;TicketID=[% Data.TicketID | html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'Pending',
  'PopupType' => 'TicketAction',
  'Target' => ''
};
```

Ticket::Frontend::MenuModule###448-Watch

Shows a link in the menu for subscribing / unsubscribing from 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'448-Watch'} = {
  'Action' => 'AgentTicketWatcher',
  'ClusterName' => '',
  'ClusterPriority' => '',
  'Description' => 'Watch this ticket',
  'Module' => 'Kernel::Output::HTML::TicketMenu::TicketWatcher',
  'Name' => 'Watch',
  'Target' => ''
};
```

Ticket::Frontend::MenuModule###450-Close

Shows a link in the menu to close 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'450-Close'} = {
  'Action' => 'AgentTicketClose',
  'ClusterName' => '',
  'ClusterPriority' => '',
  'Description' => 'Close this ticket',
  'Link' => 'Action=AgentTicketClose;TicketID=[% Data.TicketID | html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'Close',
  'PopupType' => 'TicketAction',
  'Target' => ''
};
```

Ticket::Frontend::MenuModule###460-Delete

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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'460-Delete'} = {
  'Action' => 'AgentTicketMove',
  'ClusterName' => '',
  'ClusterPriority' => '',
  'Description' => 'Delete this ticket',
  'Link' => 'Action=AgentTicketMove;TicketID=[% Data.TicketID %];DestQueue=Delete',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'Delete',
  'PopupType' => '',
  'Target' => ''
};
```

Ticket::Frontend::MenuModule###470-Junk

Shows a link to set a ticket as junk 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". To cluster menu items use for Key "ClusterName" and for the Content any name you want to see in the UI. Use "ClusterPriority" to configure the order of a certain cluster within the toolbar.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::MenuModule'}->{'470-Junk'} = {
  'Action' => 'AgentTicketMove',
  'ClusterName' => '',
  'ClusterPriority' => '',
  'Description' => 'Mark this ticket as junk!',
  'Link' => 'Action=AgentTicketMove;TicketID=[% Data.TicketID %];DestQueue=Junk',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'Spam',
  'PopupType' => '',
  'Target' => ''
};
```

Ticket → Frontend::Agent::Ticket::MenuModulePre

Ticket::Frontend::PreMenuModule###100-Lock

Shows a link in the menu to lock / unlock a ticket in the ticket overviews of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::PreMenuModule'}->{'100-Lock'} = {
  'Action' => 'AgentTicketLock',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Lock',
  'Name' => 'Lock',
  'PopupType' => '',
  'Target' => ''
};
```

Ticket::Frontend::PreMenuModule###200-Zoom

Shows a link in the menu to zoom a ticket in the ticket overviews of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::PreMenuModule'}->{'200-Zoom'} = {
  'Action' => 'AgentTicketZoom',
  'Description' => 'Look into a ticket!',
  'Link' => 'Action=AgentTicketZoom;TicketID=[% Data.TicketID | html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'Zoom',
  'PopupType' => '',
  'Target' => ''
};
```

Ticket::Frontend::PreMenuModule###210-History

Shows a link in the menu to see the history of a ticket in every ticket overview of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::PreMenuModule'}->{'210-History'} = {
  'Action' => 'AgentTicketHistory',
  'Description' => 'Show the ticket history',
  'Link' => 'Action=AgentTicketHistory;TicketID=[% Data.TicketID | html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'History',
  'PopupType' => 'TicketHistory',
  'Target' => ''
};
```

Ticket::Frontend::PreMenuModule###300-Priority

Shows a link in the menu to set the priority of a ticket in every ticket overview of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::PreMenuModule'}->{'300-Priority'} = {
  'Action' => 'AgentTicketPriority',
  'Description' => 'Change the priority for this ticket',
  'Link' => 'Action=AgentTicketPriority;TicketID=[% Data.TicketID | html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'Priority',
  'PopupType' => 'TicketAction',
  'Target' => ''
};
```

Ticket::Frontend::PreMenuModule###420-Note

Shows a link in the menu to add a note to a ticket in every ticket overview of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::PreMenuModule'}->{'420-Note'} = {
  'Action' => 'AgentTicketNote',
  'Description' => 'Add a note to this ticket',
  'Link' => 'Action=AgentTicketNote;TicketID=[% Data.TicketID | html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'Note',
  'PopupType' => 'TicketAction',
  'Target' => ''
};
```

Ticket::Frontend::PreMenuModule###440-Close

Shows a link in the menu to close a ticket in every ticket overview of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::PreMenuModule'}->{'440-Close'} = {
  'Action' => 'AgentTicketClose',
  'Description' => 'Close this ticket',
  'Link' => 'Action=AgentTicketClose;TicketID=[% Data.TicketID | html %]',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
  'Name' => 'Close',
  'PopupType' => 'TicketAction',
  'Target' => ''
};
```

Ticket::Frontend::PreMenuModule###445-Move

Shows a link in the menu to move a ticket in every ticket overview of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::PreMenuModule'}->{'445-Move'} = {
  'Action' => 'AgentTicketMove',
  'Description' => 'Change queue!',
  'Module' => 'Kernel::Output::HTML::TicketMenu::Move',
  'Name' => 'Move'
};
```

Ticket::Frontend::PreMenuModule###450-Delete

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".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::PreMenuModule'}->{'450-Delete'} = {
  'Action' => 'AgentTicketMove',
  'Description' => 'Delete this ticket',
  'Link' => 'Action=AgentTicketMove;TicketID=[% Data.TicketID %];DestQueue=Delete',
};
```

```
'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
'Name' => 'Delete',
'PopupType' => '',
'Target' => ''
};
```

Ticket::Frontend::PreMenuModule###460-Junk

Shows a link in the menu to set a ticket as junk 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".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::PreMenuModule'}->{'460-Junk'} = {
'Action' => 'AgentTicketMove',
'Description' => 'Mark as Spam!',
'Link' => 'Action=AgentTicketMove;TicketID=[% Data.TicketID %];DestQueue=Junk',
'Module' => 'Kernel::Output::HTML::TicketMenu::Generic',
'Name' => 'Spam',
'PopupType' => '',
'Target' => ''
};
```

Ticket → Frontend::Agent::Ticket::OverviewMenuModule

Ticket::Frontend::OverviewMenuModule###001-Sort

Shows a select of ticket attributes to order the queue view ticket list. The possible selections can be configured via 'TicketOverviewMenuSort###SortAttributes'.

Standardwert:

```
$Self->{'Ticket::Frontend::OverviewMenuModule'}->{'001-Sort'} = {
'Module' => 'Kernel::Output::HTML::TicketOverviewMenu::Sort'
};
```

TicketOverviewMenuSort###SortAttributes

Defines from which ticket attributes the agent can select the result order.

Standardwert:

```
$Self->{'TicketOverviewMenuSort'}->{'SortAttributes'} = {
'Age' => '1',
'Title' => '1'
};
```

Ticket → Frontend::Agent::Ticket::ViewBounce

Ticket::Frontend::AgentTicketBounce###Permission

Required permissions to use the ticket bounce screen in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'Permission'} = 'bounce';
```

Ticket::Frontend::AgentTicketBounce###RequiredLock

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).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'RequiredLock'} = '1';
```

Ticket::Frontend::AgentTicketBounce###StateDefault

Defines the default next state of a ticket after being bounced, in the ticket bounce screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'StateDefault'} = 'closed successful';
```

Ticket::Frontend::AgentTicketBounce###StateType

Defines the next state of a ticket after being bounced, in the ticket bounce screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketBounce'}->{'StateType'} = [
    'open',
    'closed'
];
```

Ticket::Frontend::BounceText

Defines the default ticket bounced notification for customer/sender in the ticket bounce screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::BounceText'} = 'Your email with ticket number
"<OTRS_TICKET>" is bounced to "<OTRS_BOUNCE_TO>". Contact this address for further
information.';
```

Ticket → Frontend::Agent::Ticket::ViewBulk

Ticket::Frontend::AgentTicketBulk###RequiredLock

Automatically lock and set owner to current Agent after selecting for an Bulk Action.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'RequiredLock'} = '1';
```

Ticket::Frontend::AgentTicketBulk###TicketType

Sets the ticket type in the ticket bulk screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'TicketType'} = '1';
```

Ticket::Frontend::AgentTicketBulk###Owner

Sets the ticket owner in the ticket bulk screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Owner'} = '1';
```

Ticket::Frontend::AgentTicketBulk###Responsible

Sets the responsible agent of the ticket in the ticket bulk screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Responsible'} = '1';
```

Ticket::Frontend::AgentTicketBulk###State

If a note is added by an agent, sets the state of a ticket in the ticket bulk screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'State'} = '1';
```

Ticket::Frontend::AgentTicketBulk###StateType

Defines the next state of a ticket after adding a note, in the ticket bulk screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'StateType'} = [  
  'open',  
  'closed',  
  'pending reminder',  
  'pending auto'  
];
```

Ticket::Frontend::AgentTicketBulk###StateDefault

Defines the default next state of a ticket after adding a note, in the ticket bulk screen of the agent interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'StateDefault'} = 'open';
```

Ticket::Frontend::AgentTicketBulk###Priority

Shows the ticket priority options in the ticket bulk screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'Priority'} = '1';
```

Ticket::Frontend::AgentTicketBulk###PriorityDefault

Defines the default ticket priority in the ticket bulk screen of the agent interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'PriorityDefault'} = '3 normal';
```

Ticket::Frontend::AgentTicketBulk###ArticleTypeDefault

Defines the default type of the note in the ticket bulk screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'ArticleTypeDefault'} = 'note-internal';
```

Ticket::Frontend::AgentTicketBulk###ArticleTypes

Specifies the different note types that will be used in the system.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketBulk'}->{'ArticleTypes'} = {  
  'note-external' => '1',  
  'note-internal' => '1',  
  'note-report' => '0'  
}
```

```
};
```

Ticket → Frontend::Agent::Ticket::ViewClose

Ticket::Frontend::AgentTicketClose###Permission

Required permissions to use the close ticket screen in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Permission'} = 'close';
```

Ticket::Frontend::AgentTicketClose###RequiredLock

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).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'RequiredLock'} = '1';
```

Ticket::Frontend::AgentTicketClose###TicketType

Sets the ticket type in the close ticket screen of the agent interface (Ticket::Type needs to be activated).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'TicketType'} = '0';
```

Ticket::Frontend::AgentTicketClose###Service

Sets the service in the close ticket screen of the agent interface (Ticket::Service needs to be activated).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Service'} = '0';
```

Ticket::Frontend::AgentTicketClose###ServiceMandatory

Sets if service must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ServiceMandatory'} = '0';
```

Ticket::Frontend::AgentTicketClose###SLAMandatory

Sets if SLA must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'SLAMandatory'} = '0';
```

Ticket::Frontend::AgentTicketClose###Queue

Sets the queue in the ticket close screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Queue'} = '0';
```

Ticket::Frontend::AgentTicketClose###QueueMandatory

Sets if queue must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'QueueMandatory'} = '0';
```

Ticket::Frontend::AgentTicketClose###Owner

Sets the ticket owner in the close ticket screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Owner'} = '0';
```

Ticket::Frontend::AgentTicketClose###OwnerMandatory

Sets if ticket owner must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'OwnerMandatory'} = '0';
```

Ticket::Frontend::AgentTicketClose###Responsible

Sets the responsible agent of the ticket in the close ticket screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Responsible'} = '0';
```

Ticket::Frontend::AgentTicketClose###ResponsibleMandatory

Sets if ticket responsible must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ResponsibleMandatory'} = '0';
```

Ticket::Frontend::AgentTicketClose###State

If a note is added by an agent, sets the state of a ticket in the close ticket screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'State'} = '1';
```

Ticket::Frontend::AgentTicketClose###StateMandatory

Sets if state must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'StateMandatory'} = '0';
```

Ticket::Frontend::AgentTicketClose###StateType

Defines the next state of a ticket after adding a note, in the close ticket screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'StateType'} = [  
  'closed'  
];
```

Ticket::Frontend::AgentTicketClose###StateDefault

Defines the default next state of a ticket after adding a note, in the close ticket screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'StateDefault'} = 'closed successful';
```

Ticket::Frontend::AgentTicketClose###Note

Allows adding notes in the close ticket screen of the agent interface. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Note'} = '1';
```

Ticket::Frontend::AgentTicketClose###NoteMandatory

Sets if note must be filled in by the agent. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'NoteMandatory'} = '1';
```

Ticket::Frontend::AgentTicketClose###Subject

Sets the default subject for notes added in the close ticket screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Subject'} = '';
```

Ticket::Frontend::AgentTicketClose###Body

Sets the default body text for notes added in the close ticket screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Body'} = '';
```

Ticket::Frontend::AgentTicketClose###InvolvedAgent

Shows a list of all the involved agents on this ticket, in the close ticket screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'InvolvedAgent'} = '0';
```

Ticket::Frontend::AgentTicketClose###InformAgent

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'InformAgent'} = '0';
```

Ticket::Frontend::AgentTicketClose###ArticleTypeDefault

Defines the default type of the note in the close ticket screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ArticleTypeDefault'} = 'note-internal';
```

Ticket::Frontend::AgentTicketClose###ArticleTypes

Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'ArticleTypes'} = {
```



```
'note-external' => '0',  
'note-internal' => '1',  
'note-report' => '0'  
};
```

Ticket::Frontend::AgentTicketClose###Priority

Shows the ticket priority options in the close ticket screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Priority'} = '0';
```

Ticket::Frontend::AgentTicketClose###PriorityDefault

Defines the default ticket priority in the close ticket screen of the agent interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'PriorityDefault'} = '3 normal';
```

Ticket::Frontend::AgentTicketClose###Title

Shows the title fields in the close ticket screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'Title'} = '0';
```

Ticket::Frontend::AgentTicketClose###HistoryType

Defines the history type for the close ticket screen action, which gets used for ticket history in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'HistoryType'} = 'AddNote';
```

Ticket::Frontend::AgentTicketClose###HistoryComment

Defines the history comment for the close ticket screen action, which gets used for ticket history in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'HistoryComment'} = '%Close';
```

Ticket::Frontend::AgentTicketClose###DynamicField

Dynamic fields shown in the ticket close screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'DynamicField'} = {};
```

Ticket::Frontend::AgentTicketClose###RichTextWidth

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'RichTextWidth'} = '620';
```

Ticket::Frontend::AgentTicketClose###RichTextHeight

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketClose'}->{'RichTextHeight'} = '100';
```

Ticket → Frontend::Agent::Ticket::ViewCompose

Ticket::Frontend::AgentTicketCompose###Permission

Required permissions to use the ticket compose screen in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'Permission'} = 'compose';
```

Ticket::Frontend::AgentTicketCompose###RequiredLock

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).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'RequiredLock'} = '1';
```

Ticket::Frontend::AgentTicketCompose###StateDefault

Defines the default next state of a ticket if it is composed / answered in the ticket compose screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'StateDefault'} = 'open';
```

Ticket::Frontend::AgentTicketCompose###StateType

Defines the next possible states after composing / answering a ticket in the ticket compose screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'StateType'} = [
  'open',
  'closed',
  'pending auto',
  'pending reminder'
];
```

Ticket::Frontend::AgentTicketCompose###ArticleTypes

Specifies the different article types that will be used in the system.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'ArticleTypes'} = [
  'email-external',
  'email-internal'
];
```

Ticket::Frontend::AgentTicketCompose###DefaultArticleType

Specifies the default article type for the ticket compose screen in the agent interface if the article type cannot be automatically detected.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'DefaultArticleType'} = 'email-external';
```

Ticket::Frontend::ResponseFormat

Defines the format of responses in the ticket compose screen of the agent interface ([% Data.OrigFrom | html %] is From 1:1, [% Data.OrigFromName | html %] is only realname of From).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ResponseFormat'} = '[% Data.Salutation | html %] [% Data.StdResponse | html %] [% Data.Signature | html %] [% Data.Created | Localize("TimeShort") %] - [% Data.OrigFromName | html %] [% Translate("wrote") | html %]: [% Data.Body | html %]';
```

Ticket::Frontend::Quote

Defines the used character for plaintext email quotes in the ticket compose screen of the agent interface. If this is empty or inactive, original emails will not be quoted but appended to the response.

Standardwert:

```
$Self->{'Ticket::Frontend::Quote'} = '>';
```

Ticket::Frontend::ResponseQuoteMaxLines

Defines the maximum number of quoted lines to be added to responses.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::ResponseQuoteMaxLines'} = '99';
```

Ticket::Frontend::ComposeAddCustomerAddress

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ComposeAddCustomerAddress'} = '1';
```

Ticket::Frontend::ComposeReplaceSenderAddress

Replaces the original sender with current customer's email address on compose answer in the ticket compose screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ComposeReplaceSenderAddress'} = '0';
```

Ticket::Frontend::AgentTicketCompose###DynamicField

Dynamic fields shown in the ticket compose screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketCompose'}->{'DynamicField'} = {};
```

Ticket → Frontend::Agent::Ticket::ViewCustomer

Ticket::Frontend::AgentTicketCustomer###Permission

Required permissions to change the customer of a ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketCustomer'}->{'Permission'} = 'customer';
```

Ticket::Frontend::AgentTicketCustomer###RequiredLock

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).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketCustomer'}->{'RequiredLock'} = '0';
```

Ticket::Frontend::AgentTicketCustomer::CustomerIDReadOnly

Controls if CustomerID is editable in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketCustomer::CustomerIDReadOnly'} = '1';
```

Ticket → Frontend::Agent::Ticket::ViewEmailNew

Ticket::Frontend::AgentTicketEmail###Priority

Sets the default priority for new email tickets in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Priority'} = '3 normal';
```

Ticket::Frontend::AgentTicketEmail###ArticleType

Sets the default article type for new email tickets in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'ArticleType'} = 'email-external';
```

Ticket::Frontend::AgentTicketEmail###SenderType

Sets the default sender type for new email tickets in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'SenderType'} = 'agent';
```

Ticket::Frontend::AgentTicketEmail::CustomerIDReadOnly

Controls if CustomerID is editable in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmail::CustomerIDReadOnly'} = '1';
```

Ticket::Frontend::AgentTicketEmail###Subject

Sets the default subject for new email tickets (e.g. 'email Outbound') in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Subject'} = '';
```

Ticket::Frontend::AgentTicketEmail###Body

Sets the default text for new email tickets in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'Body'} = '';
```

Ticket::Frontend::AgentTicketEmail###StateDefault

Sets the default next ticket state, after the creation of an email ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'StateDefault'} = 'open';
```

Ticket::Frontend::AgentTicketEmail###StateType

Determines the next possible ticket states, after the creation of a new email ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'StateType'} = [
    'open',
    'pending auto',
    'pending reminder',
    'closed'
];
```

Ticket::Frontend::AgentTicketEmail###HistoryType

Defines the history type for the email ticket screen action, which gets used for ticket history in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'HistoryType'} = 'EmailAgent';
```

Ticket::Frontend::AgentTicketEmail###HistoryComment

Defines the history comment for the email ticket screen action, which gets used for ticket history in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'HistoryComment'} = '';
```

Ticket::Frontend::AgentTicketEmail###ServiceMandatory

Sets if service must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'ServiceMandatory'} = '0';
```

Ticket::Frontend::AgentTicketEmail###SLAMandatory

Sets if SLA must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'SLAMandatory'} = '0';
```

Ticket::Frontend::AgentTicketEmail###DynamicField

Dynamic fields shown in the ticket email screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'DynamicField'} = {};
```

Ticket::Frontend::AgentTicketEmail###RichTextWidth

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'RichTextWidth'} = '620';
```

Ticket::Frontend::AgentTicketEmail###RichTextHeight

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmail'}->{'RichTextHeight'} = '320';
```

Ticket → Frontend::Agent::Ticket::ViewEmailOutbound

Ticket::Frontend::AgentTicketEmailOutbound###Permission

Required permissions to use the email outbound screen in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmailOutbound'}->{'Permission'} = 'compose';
```

Ticket::Frontend::AgentTicketEmailOutbound###RequiredLock

Defines if a ticket lock is required in the email 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).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmailOutbound'}->{'RequiredLock'} = '1';
```

Ticket::Frontend::AgentTicketEmailOutbound###StateDefault

Defines the default next state of a ticket after the message has been sent, in the email outbound screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmailOutbound'}->{'StateDefault'} = 'open';
```

Ticket::Frontend::AgentTicketEmailOutbound###StateType

Defines the next possible states after sending a message in the email outbound screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmailOutbound'}->{'StateType'} = [
  'open',
  'closed',
  'pending reminder',
  'pending auto'
];
```

Ticket::Frontend::AgentTicketEmailOutbound###ArticleTypeDefault

Defines the default type of the message in the email outbound screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmailOutbound'}->{'ArticleTypeDefault'} =
'email-internal';
```

Ticket::Frontend::AgentTicketEmailOutbound###ArticleTypes

Specifies the different article types that will be used in the system.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmailOutbound'}->{'ArticleTypes'} = [
  'email-external',
  'email-internal'
];
```

Ticket::Frontend::AgentTicketEmailOutbound###DynamicField

Dynamic fields shown in the email outbound screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmailOutbound'}->{'DynamicField'} = {};
```

Ticket::Frontend::AgentTicketEmailOutbound###RichTextWidth

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmailOutbound'}->{'RichTextWidth'} = '620';
```

Ticket::Frontend::AgentTicketEmailOutbound###RichTextHeight

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEmailOutbound'}->{'RichTextHeight'} = '300';
```

Ticket → Frontend::Agent::Ticket::ViewEscalation

Ticket::Frontend::AgentTicketEscalationView###TicketPermission

Defines the required permission to show a ticket in the escalation view of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'TicketPermission'} = 'rw';
```

Ticket::Frontend::AgentTicketEscalationView###ViewableTicketsPage

Shows all open tickets (even if they are locked) in the escalation view of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'ViewableTicketsPage'} = '50';
```

Ticket::Frontend::AgentTicketEscalationView###SortBy::Default

Defines the default ticket attribute for ticket sorting in the escalation view of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'SortBy::Default'} = 'EscalationTime';
```

Ticket::Frontend::AgentTicketEscalationView###Order::Default

Defines the default ticket order (after priority sort) in the escalation view of the agent interface. Up: oldest on top. Down: latest on top.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'Order::Default'} = 'Up';
```

Ticket::Frontend::AgentTicketEscalationView###DefaultColumns

Columns that can be filtered in the escalation view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes, Dynamic Fields (DynamicField_NameX) and Customer attributes (e.g. CustomerUser-Phone, CustomerCompanyName, ...) are allowed.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketEscalationView'}->{'DefaultColumns'} = {  
  'Age' => '2',  
  'Changed' => '1',  
  'Created' => '1',  
  'CustomerCompanyName' => '1',  
  'CustomerID' => '2',  
  'CustomerName' => '1',  
  'CustomerUserID' => '1',  
  'EscalationResponseTime' => '1',  
  'EscalationSolutionTime' => '1',  
  'EscalationTime' => '2',  
  'EscalationUpdateTime' => '1',  
  'Lock' => '2',  
  'Owner' => '2',  
  'PendingTime' => '1',  
  'Priority' => '1',  
  'Queue' => '2',  
  'Responsible' => '1',  
  'SLA' => '1',  
  'Service' => '1',  
  'State' => '2',
```



```
'TicketNumber' => '2',
'Title' => '2',
'Type' => '1'
};
```

Ticket → Frontend::Agent::Ticket::ViewForward

Ticket::Frontend::AgentTicketForward###Permission

Required permissions to use the ticket forward screen in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketForward'}->{'Permission'} = 'forward';
```

Ticket::Frontend::AgentTicketForward###RequiredLock

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).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketForward'}->{'RequiredLock'} = '1';
```

Ticket::Frontend::AgentTicketForward###StateDefault

Defines the default next state of a ticket after being forwarded, in the ticket forward screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketForward'}->{'StateDefault'} = 'closed
successful';
```

Ticket::Frontend::AgentTicketForward###StateType

Defines the next possible states after forwarding a ticket in the ticket forward screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketForward'}->{'StateType'} = [
  'open',
  'closed',
  'pending reminder',
  'pending auto'
];
```

Ticket::Frontend::AgentTicketForward###ArticleTypeDefault

Defines the default type of forwarded message in the ticket forward screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketForward'}->{'ArticleTypeDefault'} = 'email-
external';
```

Ticket::Frontend::AgentTicketForward###ArticleTypes

Specifies the different article types that will be used in the system.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketForward'}->{'ArticleTypes'} = [
  'email-external',
  'email-internal'
];
```

Ticket::Frontend::AgentTicketForward###DynamicField

Dynamic fields shown in the ticket forward screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketForward'}->{'DynamicField'} = {};
```

Ticket::Frontend::AgentTicketForward###RichTextWidth

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketForward'}->{'RichTextWidth'} = '620';
```

Ticket::Frontend::AgentTicketForward###RichTextHeight

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketForward'}->{'RichTextHeight'} = '100';
```

Ticket → Frontend::Agent::Ticket::ViewFreeText**Ticket::Frontend::AgentTicketFreeText###Permission**

Required permissions to use the ticket free text screen in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Permission'} = 'rw';
```

Ticket::Frontend::AgentTicketFreeText###RequiredLock

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).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'RequiredLock'} = '0';
```

Ticket::Frontend::AgentTicketFreeText###TicketType

Sets the ticket type in the ticket free text screen of the agent interface (Ticket::Type needs to be activated).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'TicketType'} = '1';
```

Ticket::Frontend::AgentTicketFreeText###Service

Sets the service in the ticket free text screen of the agent interface (Ticket::Service needs to be activated).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Service'} = '1';
```

Ticket::Frontend::AgentTicketFreeText###ServiceMandatory

Sets if service must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ServiceMandatory'} = '0';
```

Ticket::Frontend::AgentTicketFreeText###SLAMandatory

Sets if SLA must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'SLAMandatory'} = '0';
```

Ticket::Frontend::AgentTicketFreeText###Queue

Sets the queue in the ticket free text screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Queue'} = '0';
```

Ticket::Frontend::AgentTicketFreeText###QueueMandatory

Sets if queue must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'QueueMandatory'} = '0';
```

Ticket::Frontend::AgentTicketFreeText###Owner

Sets the ticket owner in the ticket free text screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Owner'} = '0';
```

Ticket::Frontend::AgentTicketFreeText###OwnerMandatory

Sets if ticket owner must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'OwnerMandatory'} = '0';
```

Ticket::Frontend::AgentTicketFreeText###Responsible

Sets the responsible agent of the ticket in the ticket free text screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Responsible'} = '0';
```

Ticket::Frontend::AgentTicketFreeText###ResponsibleMandatory

Sets if ticket responsible must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ResponsibleMandatory'} = '0';
```

Ticket::Frontend::AgentTicketFreeText###State

If a note is added by an agent, sets the state of a ticket in the ticket free text screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'State'} = '0';
```

Ticket::Frontend::AgentTicketFreeText###StateMandatory

Sets if state must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'StateMandatory'} = '0';
```

Ticket::Frontend::AgentTicketFreeText###StateType

Defines the next state of a ticket after adding a note, in the ticket free text screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'StateType'} = [  
  'open',  
  'closed',  
  'pending reminder',  
  'pending auto'  
];
```

Ticket::Frontend::AgentTicketFreeText###StateDefault

Defines the default next state of a ticket after adding a note, in the ticket free text screen of the agent interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'StateDefault'} = 'open';
```

Ticket::Frontend::AgentTicketFreeText###Note

Allows adding notes in the ticket free text screen of the agent interface. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Note'} = '0';
```

Ticket::Frontend::AgentTicketFreeText###NoteMandatory

Sets if note must be filled in by the agent. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'NoteMandatory'} = '0';
```

Ticket::Frontend::AgentTicketFreeText###Subject

Defines the default subject of a note in the ticket free text screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Subject'} = '';
```

Ticket::Frontend::AgentTicketFreeText###Body

Defines the default body of a note in the ticket free text screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Body'} = '';
```

Ticket::Frontend::AgentTicketFreeText###InvolvedAgent

Shows a list of all the involved agents on this ticket, in the ticket free text screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'InvolvedAgent'} = '0';
```

Ticket::Frontend::AgentTicketFreeText###InformAgent

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'InformAgent'} = '0';
```

Ticket::Frontend::AgentTicketFreeText###ArticleTypeDefault

Defines the default type of the note in the ticket free text screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ArticleTypeDefault'} = 'note-internal';
```

Ticket::Frontend::AgentTicketFreeText###ArticleTypes

Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'ArticleTypes'} = {  
  'note-external' => '1',  
  'note-internal' => '1',  
  'note-report' => '0'  
};
```

Ticket::Frontend::AgentTicketFreeText###Priority

Shows the ticket priority options in the ticket free text screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Priority'} = '0';
```

Ticket::Frontend::AgentTicketFreeText###PriorityDefault

Defines the default ticket priority in the ticket free text screen of the agent interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'PriorityDefault'} = '3 normal';
```

Ticket::Frontend::AgentTicketFreeText###Title

Shows the title field in the ticket free text screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'Title'} = '1';
```

Ticket::Frontend::AgentTicketFreeText###HistoryType

Defines the history type for the ticket free text screen action, which gets used for ticket history.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'HistoryType'} = 'AddNote';
```

Ticket::Frontend::AgentTicketFreeText###HistoryComment

Defines the history comment for the ticket free text screen action, which gets used for ticket history.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'HistoryComment'} = '%FreeText';
```

Ticket::Frontend::AgentTicketFreeText###DynamicField

Dynamic fields shown in the ticket free text screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'DynamicField'} = {};
```

Ticket::Frontend::AgentTicketFreeText###RichTextWidth

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'RichTextWidth'} = '620';
```

Ticket::Frontend::AgentTicketFreeText###RichTextHeight

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketFreeText'}->{'RichTextHeight'} = '100';
```

Ticket → Frontend::Agent::Ticket::ViewHistory

Ticket::Frontend::HistoryOrder

Shows the ticket history (reverse ordered) in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::HistoryOrder'} = 'normal';
```

Ticket::Frontend::HistoryTypes###000-Framework

Controls how to display the ticket history entries as readable values.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::HistoryTypes'}->{'000-Framework'} = {
  'AddNote' => 'Added note (%s)',
  'ArchiveFlagUpdate' => 'Archive state changed: "%s"',
  'Bounce' => 'Bounced to "%s".',
  'CustomerUpdate' => 'Updated: %s',
  'EmailAgent' => 'Email sent to customer.',
  'EmailCustomer' => 'Added email. %s',
  'EscalationResponseTimeNotifyBefore' => 'Escalation response time forewarned',
  'EscalationResponseTimeStart' => 'Escalation response time in effect',
  'EscalationResponseTimeStop' => 'Escalation response time finished',
  'EscalationSolutionTimeNotifyBefore' => 'Escalation solution time forewarned',
  'EscalationSolutionTimeStart' => 'Escalation solution time in effect',
  'EscalationSolutionTimeStop' => 'Escalation solution time finished',
  'EscalationUpdateTimeNotifyBefore' => 'Escalation update time forewarned',
  'EscalationUpdateTimeStart' => 'Escalation update time in effect',
}
```

```
'EscalationUpdateTimeStop' => 'Escalation update time finished',
'FollowUp' => 'FollowUp for [%s]. %s',
'Forward' => 'Forwarded to "%s".',
'Lock' => 'Locked ticket.',
'LoopProtection' => 'Loop-Protection! No auto-response sent to "%s".',
'Misc' => '%s',
'Move' => 'Ticket moved into Queue "%s" (%s) from Queue "%s" (%s).',
'NewTicket' => 'New Ticket [%s] created (Q=%s;P=%s;S=%s).',
'OwnerUpdate' => 'New owner is "%s" (ID=%s).',
'PhoneCallAgent' => 'Agent called customer.',
'PhoneCallCustomer' => 'Customer called us.',
'PriorityUpdate' => 'Changed priority from "%s" (%s) to "%s" (%s).',
'Remove' => '%s',
'ResponsibleUpdate' => 'New responsible is "%s" (ID=%s).',
'SLAUpdate' => 'Updated SLA to %s (ID=%s).',
'SendAgentNotification' => '"%s" notification was sent to "%s" by "%s".',
'SendAnswer' => 'Email sent to "%s".',
'SendAutoFollowUp' => 'AutoFollowUp sent to "%s".',
'SendAutoReject' => 'AutoReject sent to "%s".',
'SendAutoReply' => 'AutoReply sent to "%s".',
'SendCustomerNotification' => 'Notification sent to "%s".',
'ServiceUpdate' => 'Updated Service to %s (ID=%s).',
'SetPendingTime' => 'Updated: %s',
'StateUpdate' => 'Old: "%s" New: "%s"',
'Subscribe' => 'Added subscription for user "%s".',
'SystemRequest' => 'System Request (%s).',
'TicketDynamicFieldUpdate' => 'Updated: %s=%s;%s=%s;%s=%s;',
'TicketLinkAdd' => 'Added link to ticket "%s".',
'TicketLinkDelete' => 'Deleted link to ticket "%s".',
'TimeAccounting' => '%s time unit(s) accounted. Now total %s time unit(s).',
'TitleUpdate' => 'Title updated: Old: "%s", New: "%s"',
'TypeUpdate' => 'Updated Type to %s (ID=%s).',
'Unlock' => 'Unlocked ticket.',
'Unsubscribe' => 'Removed subscription for user "%s".',
'WebRequestCustomer' => 'Customer request via web.'
};
```

Ticket → Frontend::Agent::Ticket::ViewLocked

Ticket::Frontend::AgentTicketLockedView###SortBy::Default

Defines the default ticket attribute for ticket sorting in the locked ticket view of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketLockedView'}->{'SortBy::Default'} = 'Age';
```

Ticket::Frontend::AgentTicketLockedView###Order::Default

Defines the default ticket order in the ticket locked view of the agent interface. Up: oldest on top. Down: latest on top.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketLockedView'}->{'Order::Default'} = 'Up';
```

Ticket::Frontend::AgentTicketLockedView###DefaultColumns

Columns that can be filtered in the locked view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes, Dynamic Fields (DynamicField_NameX) and Customer attributes (e.g. CustomerUser-Phone, CustomerCompanyName, ...) are allowed.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketLockedView'}->{'DefaultColumns'} = {
```

```
'Age' => '2',
'Changed' => '1',
'Created' => '1',
'CustomerCompanyName' => '1',
'CustomerID' => '2',
'CustomerName' => '1',
'CustomerUserID' => '1',
'EscalationResponseTime' => '1',
'EscalationSolutionTime' => '1',
'EscalationTime' => '1',
'EscalationUpdateTime' => '1',
'Lock' => '2',
'Owner' => '2',
'PendingTime' => '1',
'Priority' => '1',
'Queue' => '2',
'Responsible' => '1',
'SLA' => '1',
'Service' => '1',
'State' => '2',
'TicketNumber' => '2',
'Title' => '2',
'Type' => '1'
};
```

Ticket → Frontend::Agent::Ticket::ViewMerge

Ticket::Frontend::AgentTicketMerge###Permission

Required permissions to use the ticket merge screen of a zoomed ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'Permission'} = 'rw';
```

Ticket::Frontend::AgentTicketMerge###RequiredLock

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).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'RequiredLock'} = '1';
```

Ticket::Frontend::MergeText

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::MergeText'} = 'Your email with ticket number
"<OTRS_TICKET>" is merged to "<OTRS_MERGE_TO_TICKET>";
```

Ticket::Frontend::AutomaticMergeSubject

When tickets are merged, a note will be added automatically to the ticket which is no longer active. Here you can define the subject of this note (this subject cannot be changed by the agent).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:


```
$Self->{'Ticket::Frontend::AutomaticMergeSubject'} = 'Ticket Merged';
```

Ticket::Frontend::AutomaticMergeText

When tickets are merged, a note will be added automatically to the ticket which is no longer active. Here you can define the body of this note (this text cannot be changed by the agent).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AutomaticMergeText'} = 'Merged Ticket <OTRS_TICKET> to <OTRS_MERGE_TO_TICKET>.';
```

Ticket::Frontend::AgentTicketMerge###RichTextWidth

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'RichTextWidth'} = '620';
```

Ticket::Frontend::AgentTicketMerge###RichTextHeight

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMerge'}->{'RichTextHeight'} = '100';
```

Ticket → Frontend::Agent::Ticket::ViewMove

Ticket::Frontend::MoveType

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::MoveType'} = 'form';
```

Ticket::Frontend::AgentTicketMove###RequiredLock

Automatically lock and set owner to current Agent after opening the move ticket screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMove'}->{'RequiredLock'} = '1';
```

Ticket::Frontend::AgentTicketMove###State

Allows to set a new ticket state in the move ticket screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMove'}->{'State'} = '1';
```

Ticket::Frontend::AgentTicketMove###StateMandatory

Sets if state must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMove'}->{'StateMandatory'} = '0';
```

Ticket::Frontend::AgentTicketMove###StateType

Defines the next state of a ticket after being moved to another queue, in the move ticket screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMove'}->{'StateType'} = [  
  'open',  
  'closed'  
];
```

Ticket::Frontend::AgentTicketMove###Priority

Shows the ticket priority options in the move ticket screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Priority'} = '0';
```

Ticket::Frontend::AgentTicketMove###Note

Allows adding notes in the ticket free text screen of the agent interface. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Note'} = '0';
```

Ticket::Frontend::AgentTicketMove###NoteMandatory

Sets if note must be filled in by the agent. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMove'}->{'NoteMandatory'} = '0';
```

Ticket::Frontend::AgentTicketMove###NextScreen

Determines the next screen after the ticket is moved. LastScreenOverview will return the last overview screen (e.g. search results, queueview, dashboard). TicketZoom will return to the TicketZoom.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMove'}->{'NextScreen'} = 'TicketZoom';
```

Ticket::Frontend::AgentTicketMove###Subject

Sets the default subject for notes added in the ticket move screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Subject'} = '';
```

Ticket::Frontend::AgentTicketMove###Body

Sets the default body text for notes added in the ticket move screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMove'}->{'Body'} = '';
```

Ticket::Frontend::AgentTicketMove###RichTextWidth

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMove'}->{'RichTextWidth'} = '620';
```

Ticket::Frontend::AgentTicketMove###RichTextHeight

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMove'}->{'RichTextHeight'} = '100';
```

Ticket::Frontend::AgentTicketMove###DynamicField

Dynamic fields shown in the ticket move screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketMove'}->{'DynamicField'} = {};
```

Ticket → Frontend::Agent::Ticket::ViewNote**Ticket::Frontend::AgentTicketNote###Permission**

Required permissions to use the ticket note screen in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Permission'} = 'note';
```

Ticket::Frontend::AgentTicketNote###RequiredLock

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).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'RequiredLock'} = '0';
```

Ticket::Frontend::AgentTicketNote###TicketType

Sets the ticket type in the ticket note screen of the agent interface (Ticket::Type needs to be activated).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'TicketType'} = '0';
```

Ticket::Frontend::AgentTicketNote###Service

Sets the service in the ticket note screen of the agent interface (Ticket::Service needs to be activated).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Service'} = '0';
```

Ticket::Frontend::AgentTicketNote###ServiceMandatory

Sets if service must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'ServiceMandatory'} = '0';
```

Ticket::Frontend::AgentTicketNote###SLAMandatory

Sets if SLA must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'SLAMandatory'} = '0';
```

Ticket::Frontend::AgentTicketNote###Queue

Sets the queue in the ticket note screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Queue'} = '0';
```

Ticket::Frontend::AgentTicketNote###QueueMandatory

Sets if queue must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'QueueMandatory'} = '0';
```

Ticket::Frontend::AgentTicketNote###Owner

Sets the ticket owner in the ticket note screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Owner'} = '0';
```

Ticket::Frontend::AgentTicketNote###OwnerMandatory

Sets if ticket owner must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'OwnerMandatory'} = '0';
```

Ticket::Frontend::AgentTicketNote###Responsible

Sets the responsible agent of the ticket in the ticket note screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Responsible'} = '0';
```

Ticket::Frontend::AgentTicketNote###ResponsibleMandatory

Sets if ticket responsible must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'ResponsibleMandatory'} = '0';
```

Ticket::Frontend::AgentTicketNote###State

If a note is added by an agent, sets the state of a ticket in the ticket note screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'State'} = '0';
```

Ticket::Frontend::AgentTicketNote###StateMandatory

Sets if state must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'StateMandatory'} = '0';
```

Ticket::Frontend::AgentTicketNote###StateType

Defines the next state of a ticket after adding a note, in the ticket note screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'StateType'} = [  
  'open',  
  'closed',  
  'pending reminder',  
  'pending auto'  
];
```

Ticket::Frontend::AgentTicketNote###StateDefault

Defines the default next state of a ticket after adding a note, in the ticket note screen of the agent interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'StateDefault'} = 'open';
```

Ticket::Frontend::AgentTicketNote###Note

Allows adding notes in the ticket note screen of the agent interface. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Note'} = '1';
```

Ticket::Frontend::AgentTicketNote###NoteMandatory

Sets if note must be filled in by the agent. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'NoteMandatory'} = '1';
```

Ticket::Frontend::AgentTicketNote###Subject

Sets the default subject for notes added in the ticket note screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Subject'} = '';
```

Ticket::Frontend::AgentTicketNote###Body

Sets the default body text for notes added in the ticket note screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Body'} = '';
```

Ticket::Frontend::AgentTicketNote###InvolvedAgent

Shows a list of all the involved agents on this ticket, in the ticket note screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'InvolvedAgent'} = '0';
```

Ticket::Frontend::AgentTicketNote###InformAgent

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'InformAgent'} = '0';
```

Ticket::Frontend::AgentTicketNote###ArticleTypeDefault

Defines the default type of the note in the ticket note screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'ArticleTypeDefault'} = 'note-internal';
```

Ticket::Frontend::AgentTicketNote###ArticleTypes

Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'ArticleTypes'} = {  
  'note-external' => '1',  
  'note-internal' => '1',  
  'note-report' => '0'  
};
```

Ticket::Frontend::AgentTicketNote###Priority

Shows the ticket priority options in the ticket note screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Priority'} = '0';
```

Ticket::Frontend::AgentTicketNote###PriorityDefault

Defines the default ticket priority in the ticket note screen of the agent interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'PriorityDefault'} = '3 normal';
```

Ticket::Frontend::AgentTicketNote###Title

Shows the title fields in the ticket note screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'Title'} = '0';
```

Ticket::Frontend::AgentTicketNote###HistoryType

Defines the history type for the ticket note screen action, which gets used for ticket history in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'HistoryType'} = 'AddNote';
```

Ticket::Frontend::AgentTicketNote###HistoryComment

Defines the history comment for the ticket note screen action, which gets used for ticket history in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'HistoryComment'} = '%Note';
```

Ticket::Frontend::AgentTicketNote###DynamicField

Dynamic fields shown in the ticket note screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'DynamicField'} = {};
```

Ticket::Frontend::AgentTicketNote###RichTextWidth

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'RichTextWidth'} = '620';
```

Ticket::Frontend::AgentTicketNote###RichTextHeight

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketNote'}->{'RichTextHeight'} = '100';
```

Ticket → Frontend::Agent::Ticket::ViewOwner

Ticket::Frontend::AgentTicketOwner###Permission

Required permissions to use the ticket owner screen of a zoomed ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Permission'} = 'owner';
```

Ticket::Frontend::AgentTicketOwner###RequiredLock

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).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'RequiredLock'} = '0';
```

Ticket::Frontend::AgentTicketOwner###TicketType

Sets the ticket type in the ticket owner screen of a zoomed ticket in the agent interface (Ticket::Type needs to be activated).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'TicketType'} = '0';
```

Ticket::Frontend::AgentTicketOwner###Service

Sets the service in the ticket owner screen of a zoomed ticket in the agent interface (Ticket::Service needs to be activated).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Service'} = '0';
```

Ticket::Frontend::AgentTicketOwner###ServiceMandatory

Sets if service must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'ServiceMandatory'} = '0';
```

Ticket::Frontend::AgentTicketOwner###SLAMandatory

Sets if SLA must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'SLAMandatory'} = '0';
```

Ticket::Frontend::AgentTicketOwner###Queue

Sets the queue in the ticket owner screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Queue'} = '0';
```

Ticket::Frontend::AgentTicketOwner###QueueMandatory

Sets if queue must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'QueueMandatory'} = '0';
```

Ticket::Frontend::AgentTicketOwner###Owner

Sets the ticket owner in the ticket owner screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Owner'} = '1';
```

Ticket::Frontend::AgentTicketOwner###OwnerMandatory

Sets if ticket owner must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'OwnerMandatory'} = '1';
```

Ticket::Frontend::AgentTicketOwner###Responsible

Sets the responsible agent of the ticket in the ticket owner screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Responsible'} = '0';
```

Ticket::Frontend::AgentTicketOwner###ResponsibleMandatory

Sets if ticket responsible must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'ResponsibleMandatory'} = '0';
```

Ticket::Frontend::AgentTicketOwner###State

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.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'State'} = '0';
```

Ticket::Frontend::AgentTicketOwner###StateMandatory

Sets if state must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'StateMandatory'} = '0';
```

Ticket::Frontend::AgentTicketOwner###StateType

Defines the next state of a ticket after adding a note, in the ticket owner screen of a zoomed ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'StateType'} = [  
  'open',  
  'pending reminder',  
  'pending auto'  
];
```

Ticket::Frontend::AgentTicketOwner###StateDefault

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.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'StateDefault'} = 'open';
```

Ticket::Frontend::AgentTicketOwner###Note

Allows adding notes in the ticket owner screen of a zoomed ticket in the agent interface. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Note'} = '1';
```

Ticket::Frontend::AgentTicketOwner###NoteMandatory

Sets if note must be filled in by the agent. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'NoteMandatory'} = '1';
```

Ticket::Frontend::AgentTicketOwner###Subject

Sets the default subject for notes added in the ticket owner screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Subject'} = '';
```

Ticket::Frontend::AgentTicketOwner###Body

Sets the default body text for notes added in the ticket owner screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Body'} = '';
```

Ticket::Frontend::AgentTicketOwner###InvolvedAgent

Shows a list of all the involved agents on this ticket, in the ticket owner screen of a zoomed ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'InvolvedAgent'} = '0';
```

Ticket::Frontend::AgentTicketOwner###InformAgent

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'InformAgent'} = '0';
```

Ticket::Frontend::AgentTicketOwner###ArticleTypeDefault

Defines the default type of the note in the ticket owner screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'ArticleTypeDefault'} = 'note-internal';
```

Ticket::Frontend::AgentTicketOwner###ArticleTypes

Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'ArticleTypes'} = {  
  'note-external' => '0',  
  'note-internal' => '1',  
  'note-report' => '0'  
};
```

Ticket::Frontend::AgentTicketOwner###Priority

Shows the ticket priority options in the ticket owner screen of a zoomed ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Priority'} = '0';
```

Ticket::Frontend::AgentTicketOwner###PriorityDefault

Defines the default ticket priority in the ticket owner screen of a zoomed ticket in the agent interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'PriorityDefault'} = '3 normal';
```

Ticket::Frontend::AgentTicketOwner###Title

Shows the title fields in the ticket owner screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'Title'} = '0';
```

Ticket::Frontend::AgentTicketOwner###HistoryType

Defines the history type for the ticket owner screen action, which gets used for ticket history in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'HistoryType'} = 'AddNote';
```

Ticket::Frontend::AgentTicketOwner###HistoryComment

Defines the history comment for the ticket owner screen action, which gets used for ticket history in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'HistoryComment'} = '%Owner';
```

Ticket::Frontend::AgentTicketOwner###DynamicField

Dynamic fields shown in the ticket owner screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'DynamicField'} = {};
```

Ticket::Frontend::AgentTicketOwner###RichTextWidth

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'RichTextWidth'} = '620';
```

Ticket::Frontend::AgentTicketOwner###RichTextHeight

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketOwner'}->{'RichTextHeight'} = '100';
```

Ticket → Frontend::Agent::Ticket::ViewPending

Ticket::Frontend::AgentTicketPending###Permission

Required permissions to use the ticket pending screen of a zoomed ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Permission'} = 'pending';
```

Ticket::Frontend::AgentTicketPending###RequiredLock

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).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'RequiredLock'} = '1';
```

Ticket::Frontend::AgentTicketPending###TicketType

Sets the ticket type in the ticket pending screen of a zoomed ticket in the agent interface (Ticket::Type needs to be activated).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'TicketType'} = '0';
```

Ticket::Frontend::AgentTicketPending###Service

Sets the service in the ticket pending screen of a zoomed ticket in the agent interface (Ticket::Service needs to be activated).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Service'} = '0';
```

Ticket::Frontend::AgentTicketPending###ServiceMandatory

Sets if service must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'ServiceMandatory'} = '0';
```

Ticket::Frontend::AgentTicketPending###SLAMandatory

Sets if SLA must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'SLAMandatory'} = '0';
```

Ticket::Frontend::AgentTicketPending###Queue

Sets the queue in the ticket pending screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Queue'} = '0';
```

Ticket::Frontend::AgentTicketPending###QueueMandatory

Sets if queue must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'QueueMandatory'} = '0';
```

Ticket::Frontend::AgentTicketPending###Owner

Sets the ticket owner in the ticket pending screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Owner'} = '0';
```

Ticket::Frontend::AgentTicketPending###OwnerMandatory

Sets if ticket owner must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'OwnerMandatory'} = '0';
```

Ticket::Frontend::AgentTicketPending###Responsible

Sets the responsible agent of the ticket in the ticket pending screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Responsible'} = '0';
```

Ticket::Frontend::AgentTicketPending###ResponsibleMandatory

Sets if ticket responsible must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'ResponsibleMandatory'} = '0';
```

Ticket::Frontend::AgentTicketPending###State

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.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'State'} = '1';
```

Ticket::Frontend::AgentTicketPending###StateMandatory

Sets if state must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'StateMandatory'} = '0';
```

Ticket::Frontend::AgentTicketPending###StateType

Defines the next state of a ticket after adding a note, in the ticket pending screen of a zoomed ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'StateType'} = [  
  'pending reminder',  
  'pending auto'  
];
```

Ticket::Frontend::AgentTicketPending###StateDefault

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.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'StateDefault'} = 'pending  
reminder';
```

Ticket::Frontend::AgentTicketPending###Note

Allows adding notes in the ticket pending screen of a zoomed ticket in the agent interface. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Note'} = '1';
```

Ticket::Frontend::AgentTicketPending###NoteMandatory

Sets if note must be filled in by the agent. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'NoteMandatory'} = '1';
```

Ticket::Frontend::AgentTicketPending###Subject

Sets the default subject for notes added in the ticket pending screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Subject'} = '';
```

Ticket::Frontend::AgentTicketPending###Body

Sets the default body text for notes added in the ticket pending screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Body'} = '';
```

Ticket::Frontend::AgentTicketPending###InvolvedAgent

Shows a list of all the involved agents on this ticket, in the ticket pending screen of a zoomed ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'InvolvedAgent'} = '0';
```

Ticket::Frontend::AgentTicketPending###InformAgent

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'InformAgent'} = '0';
```

Ticket::Frontend::AgentTicketPending###ArticleTypeDefault

Defines the default type of the note in the ticket pending screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'ArticleTypeDefault'} = 'note-internal';
```

Ticket::Frontend::AgentTicketPending###ArticleTypes

Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'ArticleTypes'} = {
  'note-external' => '0',
  'note-internal' => '1',
  'note-report' => '0'
};
```

Ticket::Frontend::AgentTicketPending###Priority

Shows the ticket priority options in the ticket pending screen of a zoomed ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Priority'} = '0';
```

Ticket::Frontend::AgentTicketPending###PriorityDefault

Defines the default ticket priority in the ticket pending screen of a zoomed ticket in the agent interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'PriorityDefault'} = '3 normal';
```

Ticket::Frontend::AgentTicketPending###Title

Shows the title fields in the ticket pending screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'Title'} = '0';
```

Ticket::Frontend::AgentTicketPending###HistoryType

Defines the history type for the ticket pending screen action, which gets used for ticket history in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'HistoryType'} = 'AddNote';
```

Ticket::Frontend::AgentTicketPending###HistoryComment

Defines the history comment for the ticket pending screen action, which gets used for ticket history in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'HistoryComment'} = '%Pending';
```

Ticket::Frontend::AgentTicketPending###DynamicField

Dynamic fields shown in the ticket pending screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'DynamicField'} = {};
```

Ticket::Frontend::AgentTicketPending###RichTextWidth

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'RichTextWidth'} = '620';
```

Ticket::Frontend::AgentTicketPending###RichTextHeight

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPending'}->{'RichTextHeight'} = '100';
```

Ticket → Frontend::Agent::Ticket::ViewPhoneInbound

Ticket::Frontend::AgentTicketPhoneInbound###Permission

Required permissions to use the ticket phone inbound screen in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'Permission'} = 'phone';
```

Ticket::Frontend::AgentTicketPhoneInbound###RequiredLock

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).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'RequiredLock'} = '0';
```

Ticket::Frontend::AgentTicketPhoneInbound###ArticleType

Defines the default type of the note in the ticket phone inbound screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'ArticleType'} = 'phone';
```

Ticket::Frontend::AgentTicketPhoneInbound###SenderType

Defines the default sender type for phone tickets in the ticket phone inbound screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'SenderType'} = 'customer';
```

Ticket::Frontend::AgentTicketPhoneInbound###Subject

Defines the default subject for phone tickets in the ticket phone inbound screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'Subject'} = '';
```

Ticket::Frontend::AgentTicketPhoneInbound###Body

Defines the default note body text for phone tickets in the ticket phone inbound screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'Body'} = '';
```

Ticket::Frontend::AgentTicketPhoneInbound###State

Defines the default ticket next state after adding a phone note in the ticket phone inbound screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'State'} = 'open';
```

Ticket::Frontend::AgentTicketPhoneInbound###StateType

Next possible ticket states after adding a phone note in the ticket phone inbound screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:


```
$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'StateType'} = [
  'open',
  'pending auto',
  'pending reminder',
  'closed'
];
```

Ticket::Frontend::AgentTicketPhoneInbound###HistoryType

Defines the history type for the ticket phone inbound screen action, which gets used for ticket history in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'HistoryType'} =
  'PhoneCallCustomer';
```

Ticket::Frontend::AgentTicketPhoneInbound###HistoryComment

Defines the history comment for the ticket phone inbound screen action, which gets used for ticket history in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'HistoryComment'} = '';
```

Ticket::Frontend::AgentTicketPhoneInbound###DynamicField

Dynamic fields shown in the ticket phone inbound screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'DynamicField'} = {};
```

Ticket::Frontend::AgentTicketPhoneInbound###RichTextWidth

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'RichTextWidth'} = '475';
```

Ticket::Frontend::AgentTicketPhoneInbound###RichTextHeight

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneInbound'}->{'RichTextHeight'} = '200';
```

Ticket → Frontend::Agent::Ticket::ViewPhoneNew

Ticket::Frontend::AgentTicketPhone###Priority

Sets the default priority for new phone tickets in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'Priority'} = '3 normal';
```

Ticket::Frontend::AgentTicketPhone###ArticleType

Sets the default article type for new phone tickets in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'ArticleType'} = 'phone';
```

Ticket::Frontend::AgentTicketPhone###SenderType

Sets the default sender type for new phone ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'SenderType'} = 'customer';
```

Ticket::Frontend::AgentTicketPhone::CustomerIDReadOnly

Controls if CustomerID is editable in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone::CustomerIDReadOnly'} = '1';
```

Ticket::Frontend::AgentTicketPhone::AllowMultipleFrom

Controls if more than one from entry can be set in the new phone ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone::AllowMultipleFrom'} = '1';
```

Ticket::Frontend::AgentTicketPhone###Subject

Sets the default subject for new phone tickets (e.g. 'Phone call') in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'Subject'} = '';
```

Ticket::Frontend::AgentTicketPhone###Body

Sets the default note text for new telephone tickets. E.g 'New ticket via call' in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'Body'} = '';
```

Ticket::Frontend::AgentTicketPhone###StateDefault

Sets the default next state for new phone tickets in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'StateDefault'} = 'open';
```

Ticket::Frontend::AgentTicketPhone###StateType

Determines the next possible ticket states, after the creation of a new phone ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'StateType'} = [  
  'open',  
  'pending auto',  
  'pending reminder',  
  'closed'  
];
```

Ticket::Frontend::AgentTicketPhone###HistoryType

Defines the history type for the phone ticket screen action, which gets used for ticket history in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'HistoryType'} = 'PhoneCallCustomer';
```

Ticket::Frontend::AgentTicketPhone###HistoryComment

Defines the history comment for the phone ticket screen action, which gets used for ticket history in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'HistoryComment'} = '';
```

Ticket::Frontend::AgentTicketPhone###SplitLinkType

Sets the default link type of splitted tickets in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'SplitLinkType'} = {  
  'Direction' => 'Target',  
  'LinkType' => 'ParentChild'  
};
```

Ticket::Frontend::AgentTicketPhone###ServiceMandatory

Sets if service must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'ServiceMandatory'} = '0';
```

Ticket::Frontend::AgentTicketPhone###SLAMandatory

Sets if SLA must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'SLAMandatory'} = '0';
```

Ticket::Frontend::AgentTicketPhone###DynamicField

Dynamic fields shown in the ticket phone screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'DynamicField'} = {};
```

Ticket::Frontend::AgentTicketPhone###RichTextWidth

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'RichTextWidth'} = '620';
```

Ticket::Frontend::AgentTicketPhone###RichTextHeight

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhone'}->{'RichTextHeight'} = '320';
```

Ticket → Frontend::Agent::Ticket::ViewPhoneOutbound

Ticket::Frontend::AgentTicketPhoneOutbound###Permission

Required permissions to use the ticket phone outbound screen in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'Permission'} = 'phone';
```

Ticket::Frontend::AgentTicketPhoneOutbound###RequiredLock

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).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'RequiredLock'} = '1';
```

Ticket::Frontend::AgentTicketPhoneOutbound###ArticleType

Defines the default type of the note in the ticket phone outbound screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'ArticleType'} = 'phone';
```

Ticket::Frontend::AgentTicketPhoneOutbound###SenderType

Defines the default sender type for phone tickets in the ticket phone outbound screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'SenderType'} = 'agent';
```

Ticket::Frontend::AgentTicketPhoneOutbound###Subject

Defines the default subject for phone tickets in the ticket phone outbound screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'Subject'} = '';
```

Ticket::Frontend::AgentTicketPhoneOutbound###Body

Defines the default note body text for phone tickets in the ticket phone outbound screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'Body'} = '';
```

Ticket::Frontend::AgentTicketPhoneOutbound###State

Defines the default ticket next state after adding a phone note in the ticket phone outbound screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'State'} = 'closed  
successful';
```

Ticket::Frontend::AgentTicketPhoneOutbound###StateType

Next possible ticket states after adding a phone note in the ticket phone outbound screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'StateType'} = [  
  'open',  
  'pending auto',  
  'pending reminder',  
  'closed'  
];
```

Ticket::Frontend::AgentTicketPhoneOutbound###HistoryType

Defines the history type for the ticket phone outbound screen action, which gets used for ticket history in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'HistoryType'} =  
'PhoneCallAgent';
```

Ticket::Frontend::AgentTicketPhoneOutbound###HistoryComment

Defines the history comment for the ticket phone outbound screen action, which gets used for ticket history in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'HistoryComment'} = '';
```

Ticket::Frontend::AgentTicketPhoneOutbound###DynamicField

Dynamic fields shown in the ticket phone outbound screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'DynamicField'} = {};
```

Ticket::Frontend::AgentTicketPhoneOutbound###RichTextWidth

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'RichTextWidth'} = '475';
```

Ticket::Frontend::AgentTicketPhoneOutbound###RichTextHeight

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPhoneOutbound'}->{'RichTextHeight'} = '200';
```

Ticket → Frontend::Agent::Ticket::ViewPrint

Ticket::Frontend::AgentTicketPrint###DynamicField

Dynamic fields shown in the ticket print screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPrint'}->{'DynamicField'} = {};
```

Ticket → Frontend::Agent::Ticket::ViewPriority

Ticket::Frontend::AgentTicketPriority###Permission

Required permissions to use the ticket priority screen of a zoomed ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Permission'} = 'priority';
```

Ticket::Frontend::AgentTicketPriority###RequiredLock

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).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'RequiredLock'} = '1';
```

Ticket::Frontend::AgentTicketPriority###TicketType

Sets the ticket type in the ticket priority screen of a zoomed ticket in the agent interface (Ticket::Type needs to be activated).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'TicketType'} = '0';
```

Ticket::Frontend::AgentTicketPriority###Service

Sets the service in the ticket priority screen of a zoomed ticket in the agent interface (Ticket::Service needs to be activated).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Service'} = '0';
```

Ticket::Frontend::AgentTicketPriority###ServiceMandatory

Sets if service must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ServiceMandatory'} = '0';
```

Ticket::Frontend::AgentTicketPriority###SLAMandatory

Sets if SLA must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'SLAMandatory'} = '0';
```

Ticket::Frontend::AgentTicketPriority###Queue

Sets the queue in the ticket priority screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Queue'} = '0';
```

Ticket::Frontend::AgentTicketPriority###QueueMandatory

Sets if queue must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'QueueMandatory'} = '0';
```

Ticket::Frontend::AgentTicketPriority###Owner

Sets the ticket owner in the ticket priority screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Owner'} = '0';
```

Ticket::Frontend::AgentTicketPriority###OwnerMandatory

Sets if ticket owner must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'OwnerMandatory'} = '0';
```

Ticket::Frontend::AgentTicketPriority###Responsible

Sets the responsible agent of the ticket in the ticket priority screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Responsible'} = '0';
```

Ticket::Frontend::AgentTicketPriority###ResponsibleMandatory

Sets if ticket responsible must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ResponsibleMandatory'} = '0';
```

Ticket::Frontend::AgentTicketPriority###State

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.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'State'} = '0';
```

Ticket::Frontend::AgentTicketPriority###StateMandatory

Sets if state must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'StateMandatory'} = '0';
```

Ticket::Frontend::AgentTicketPriority###StateType

Defines the next state of a ticket after adding a note, in the ticket priority screen of a zoomed ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'StateType'} = [  
  'open',  
  'pending reminder',  
  'pending auto'  
];
```

Ticket::Frontend::AgentTicketPriority###StateDefault

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.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'StateDefault'} = 'open';
```

Ticket::Frontend::AgentTicketPriority###Note

Allows adding notes in the ticket priority screen of a zoomed ticket in the agent interface. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Note'} = '1';
```

Ticket::Frontend::AgentTicketPriority###NoteMandatory

Sets if note must be filled in by the agent. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'NoteMandatory'} = '1';
```

Ticket::Frontend::AgentTicketPriority###Subject

Sets the default subject for notes added in the ticket priority screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Subject'} = '';
```

Ticket::Frontend::AgentTicketPriority###Body

Sets the default body text for notes added in the ticket priority screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Body'} = '';
```

Ticket::Frontend::AgentTicketPriority###InvolvedAgent

Shows a list of all the involved agents on this ticket, in the ticket priority screen of a zoomed ticket in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'InvolvedAgent'} = '0';
```


Ticket::Frontend::AgentTicketPriority###InformAgent

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'InformAgent'} = '0';
```

Ticket::Frontend::AgentTicketPriority###ArticleTypeDefault

Defines the default type of the note in the ticket priority screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ArticleTypeDefault'} = 'note-internal';
```

Ticket::Frontend::AgentTicketPriority###ArticleTypes

Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'ArticleTypes'} = {  
  'note-external' => '0',  
  'note-internal' => '1',  
  'note-report' => '0'  
};
```

Ticket::Frontend::AgentTicketPriority###Priority

Shows the ticket priority options in the ticket priority screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Priority'} = '1';
```

Ticket::Frontend::AgentTicketPriority###PriorityDefault

Defines the default ticket priority in the ticket priority screen of a zoomed ticket in the agent interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'PriorityDefault'} = '3 normal';
```

Ticket::Frontend::AgentTicketPriority###Title

Shows the title fields in the ticket priority screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'Title'} = '0';
```

Ticket::Frontend::AgentTicketPriority###HistoryType

Defines the history type for the ticket priority screen action, which gets used for ticket history in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'HistoryType'} = 'AddNote';
```

Ticket::Frontend::AgentTicketPriority###HistoryComment

Defines the history comment for the ticket priority screen action, which gets used for ticket history in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'HistoryComment'} = '%Priority';
```

Ticket::Frontend::AgentTicketPriority###DynamicField

Dynamic fields shown in the ticket priority screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'DynamicField'} = {};
```

Ticket::Frontend::AgentTicketPriority###RichTextWidth

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'RichTextWidth'} = '620';
```

Ticket::Frontend::AgentTicketPriority###RichTextHeight

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketPriority'}->{'RichTextHeight'} = '100';
```

Ticket → Frontend::Agent::Ticket::ViewQueue

Ticket::Frontend::AgentTicketQueue###StripEmptyLines

Strips empty lines on the ticket preview in the queue view.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'StripEmptyLines'} = '0';
```

Ticket::Frontend::AgentTicketQueue###ViewAllPossibleTickets

Shows all both ro and rw queues in the queue view.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'ViewAllPossibleTickets'} = '0';
```

Ticket::Frontend::AgentTicketQueue###HideEmptyQueues

Show queues even when only locked tickets are in.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'HideEmptyQueues'} = '0';
```

Ticket::Frontend::AgentTicketQueue###HighlightAge1

Sets the age in minutes (first level) for highlighting queues that contain untouched tickets.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'HighlightAge1'} = '1440';
```

Ticket::Frontend::AgentTicketQueue###HighlightAge2

Sets the age in minutes (second level) for highlighting queues that contain untouched tickets.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'HighlightAge2'} = '2880';
```

Ticket::Frontend::AgentTicketQueue###Blink

Activates a blinking mechanism of the queue that contains the oldest ticket.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'Blink'} = '1';
```

Ticket::Frontend::AgentTicketQueue###UseSubQueues

Include tickets of subqueues per default when selecting a queue.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'UseSubQueues'} = '0';
```

Ticket::Frontend::AgentTicketQueue###QueueSort

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'QueueSort'} = {  
  '3' => '0',  
  '7' => '1'  
};
```

Ticket::Frontend::AgentTicketQueue###SortBy::Default

Defines the default sort criteria for all queues displayed in the queue view.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'SortBy::Default'} = 'Age';
```

Ticket::Frontend::AgentTicketQueue###PreSort::ByPriority

Defines if a pre-sorting by priority should be done in the queue view.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'PreSort::ByPriority'} = '1';
```

Ticket::Frontend::AgentTicketQueue###Order::Default

Defines the default sort order for all queues in the queue view, after priority sort.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'Order::Default'} = 'Up';
```

Ticket::Frontend::AgentTicketQueue###DefaultColumns

Columns that can be filtered in the queue view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes, Dynamic Fields (DynamicField_NameX) and Customer attributes (e.g. CustomerUserID, CustomerName, CustomerCompany, ...) are allowed.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketQueue'}->{'DefaultColumns'} = {
  'Age' => '2',
  'Changed' => '1',
  'Created' => '1',
  'CustomerCompanyName' => '1',
  'CustomerID' => '2',
  'CustomerName' => '1',
  'CustomerUserID' => '1',
  'EscalationResponseTime' => '1',
  'EscalationSolutionTime' => '1',
  'EscalationTime' => '1',
  'EscalationUpdateTime' => '1',
  'Lock' => '2',
  'Owner' => '2',
  'PendingTime' => '1',
  'Priority' => '1',
  'Queue' => '2',
  'Responsible' => '1',
  'SLA' => '1',
  'Service' => '1',
  'State' => '2',
  'TicketNumber' => '2',
  'Title' => '2',
  'Type' => '1'
};
```

Ticket → Frontend::Agent::Ticket::ViewResponsible

Ticket::Frontend::AgentTicketResponsibleView###SortBy::Default

Defines the default ticket attribute for ticket sorting in the responsible view of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsibleView'}->{'SortBy::Default'} = 'Age';
```

Ticket::Frontend::AgentTicketResponsibleView###Order::Default

Defines the default ticket order in the responsible view of the agent interface. Up: oldest on top. Down: latest on top.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsibleView'}->{'Order::Default'} = 'Up';
```

Ticket::Frontend::AgentTicketResponsible###Permission

Required permissions to use the ticket responsible screen in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Permission'} = 'responsible';
```

Ticket::Frontend::AgentTicketResponsible###RequiredLock

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).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'RequiredLock'} = '0';
```

Ticket::Frontend::AgentTicketResponsible###TicketType

Sets the ticket type in the ticket responsible screen of the agent interface (Ticket::Type needs to be activated).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'TicketType'} = '0';
```

Ticket::Frontend::AgentTicketResponsible###Service

Sets the service in the ticket responsible screen of the agent interface (Ticket::Service needs to be activated).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Service'} = '0';
```

Ticket::Frontend::AgentTicketResponsible###ServiceMandatory

Sets if service must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'ServiceMandatory'} = '0';
```

Ticket::Frontend::AgentTicketResponsible###SLAMandatory

Sets if SLA must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'SLAMandatory'} = '0';
```

Ticket::Frontend::AgentTicketResponsible###Queue

Sets the queue in the ticket responsible screen of a zoomed ticket in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Queue'} = '0';
```

Ticket::Frontend::AgentTicketResponsible###QueueMandatory

Sets if queue must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'QueueMandatory'} = '0';
```

Ticket::Frontend::AgentTicketResponsible###Owner

Sets the ticket owner in the ticket responsible screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Owner'} = '0';
```

Ticket::Frontend::AgentTicketResponsible###OwnerMandatory

Sets if ticket owner must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'OwnerMandatory'} = '0';
```

Ticket::Frontend::AgentTicketResponsible###Responsible

Sets the responsible agent of the ticket in the ticket responsible screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Responsible'} = '1';
```

Ticket::Frontend::AgentTicketResponsible###ResponsibleMandatory

Sets if ticket responsible must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'ResponsibleMandatory'} = '1';
```

Ticket::Frontend::AgentTicketResponsible###State

If a note is added by an agent, sets the state of a ticket in the ticket responsible screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'State'} = '0';
```

Ticket::Frontend::AgentTicketResponsible###StateMandatory

Sets if state must be selected by the agent.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'StateMandatory'} = '0';
```

Ticket::Frontend::AgentTicketResponsible###StateType

Defines the next state of a ticket after adding a note, in the ticket responsible screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'StateType'} = [  
  'open',  
  'pending reminder',  
  'pending auto'  
];
```

Ticket::Frontend::AgentTicketResponsible###StateDefault

Defines the default next state of a ticket after adding a note, in the ticket responsible screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'StateDefault'} = 'open';
```

Ticket::Frontend::AgentTicketResponsible###Note

Allows adding notes in the ticket responsible screen of the agent interface. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Note'} = '1';
```

Ticket::Frontend::AgentTicketResponsible###NoteMandatory

Sets if note must be filled in by the agent. Can be overwritten by Ticket::Frontend::NeedAccountedTime.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'NoteMandatory'} = '1';
```

Ticket::Frontend::AgentTicketResponsible###Subject

Sets the default subject for notes added in the ticket responsible screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Subject'} = '';
```

Ticket::Frontend::AgentTicketResponsible###Body

Sets the default body text for notes added in the ticket responsible screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Body'} = '';
```

Ticket::Frontend::AgentTicketResponsible###InvolvedAgent

Shows a list of all the involved agents on this ticket, in the ticket responsible screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'InvolvedAgent'} = '0';
```

Ticket::Frontend::AgentTicketResponsible###InformAgent

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.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'InformAgent'} = '0';
```

Ticket::Frontend::AgentTicketResponsible###ArticleTypeDefault

Defines the default type of the note in the ticket responsible screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'ArticleTypeDefault'} = 'note-internal';
```

Ticket::Frontend::AgentTicketResponsible###ArticleTypes

Specifies the available note types for this ticket mask. If the option is deselected, ArticleTypeDefault is used and the option is removed from the mask.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'ArticleTypes'} = {
  'note-external' => '0',
  'note-internal' => '1',
  'note-report' => '0'
};
```

Ticket::Frontend::AgentTicketResponsible###Priority

Shows the ticket priority options in the ticket responsible screen of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Priority'} = '0';
```

Ticket::Frontend::AgentTicketResponsible###PriorityDefault

Defines the default ticket priority in the ticket responsible screen of the agent interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'PriorityDefault'} = '3 normal';
```

Ticket::Frontend::AgentTicketResponsible###Title

Shows the title fields in the ticket responsible screen of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'Title'} = '1';
```

Ticket::Frontend::AgentTicketResponsible###HistoryType

Defines the history type for the ticket responsible screen action, which gets used for ticket history in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'HistoryType'} = 'AddNote';
```

Ticket::Frontend::AgentTicketResponsible###HistoryComment

Defines the history comment for the ticket responsible screen action, which gets used for ticket history in the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'HistoryComment'} = '%  
%Responsible';
```

Ticket::Frontend::AgentTicketResponsible###DynamicField

Dynamic fields shown in the ticket responsible screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'DynamicField'} = {};
```

Ticket::Frontend::AgentTicketResponsible###RichTextWidth

Defines the width for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'RichTextWidth'} = '620';
```

Ticket::Frontend::AgentTicketResponsible###RichTextHeight

Defines the height for the rich text editor component for this screen. Enter number (pixels) or percent value (relative).

Standardwert:


```
$Self->{'Ticket::Frontend::AgentTicketResponsible'}->{'RichTextHeight'} = '100';
```

Ticket::Frontend::AgentTicketResponsibleView###DefaultColumns

Columns that can be filtered in the responsible view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes, Dynamic Fields (DynamicField_NameX) and Customer attributes (e.g. CustomerUserPhone, CustomerCompanyName, ...) are allowed.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketResponsibleView'}->{'DefaultColumns'} = {
  'Age' => '2',
  'Changed' => '1',
  'Created' => '1',
  'CustomerCompanyName' => '1',
  'CustomerID' => '2',
  'CustomerName' => '1',
  'CustomerUserID' => '1',
  'EscalationResponseTime' => '1',
  'EscalationSolutionTime' => '1',
  'EscalationTime' => '1',
  'EscalationUpdateTime' => '1',
  'Lock' => '2',
  'Owner' => '2',
  'PendingTime' => '1',
  'Priority' => '1',
  'Queue' => '2',
  'Responsible' => '1',
  'SLA' => '1',
  'Service' => '1',
  'State' => '2',
  'TicketNumber' => '2',
  'Title' => '2',
  'Type' => '1'
};
```

Ticket → Frontend::Agent::Ticket::ViewSearch

Ticket::Frontend::AgentTicketSearch###ExtendedSearchCondition

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)".

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'ExtendedSearchCondition'} = '1';
```

Ticket::Frontend::AgentTicketSearch###SearchLimit

Maximum number of tickets to be displayed in the result of a search in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchLimit'} = '2000';
```

Ticket::Frontend::AgentTicketSearch###SearchPageShown

Number of tickets to be displayed in each page of a search result in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchPageShown'} = '40';
```

Ticket::Frontend::AgentTicketSearch###SearchViewableTicketLines

Number of lines (per ticket) that are shown by the search utility in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchViewableTicketLines'} = '10';
```

Ticket::Frontend::AgentTicketSearch###SortBy::Default

Defines the default ticket attribute for ticket sorting of the ticket search result of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SortBy::Default'} = 'Age';
```

Ticket::Frontend::AgentTicketSearch###Order::Default

Defines the default ticket order in the ticket search result of the agent interface. Up: oldest on top. Down: latest on top.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Order::Default'} = 'Down';
```

Ticket::Frontend::AgentTicketSearch###SearchArticleCSVTree

Exports the whole article tree in search result (it can affect the system performance).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchArticleCSVTree'} = '0';
```

Ticket::Frontend::AgentTicketSearch###SearchCSVData

Data used to export the search result in CSV format.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchCSVData'} = [  
  'TicketNumber',  
  'Age',  
  'Created',  
  'Closed',  
  'FirstLock',  
  'FirstResponse',  
  'State',  
  'Priority',  
  'Queue',  
  'Lock',  
  'Owner',  
  'UserFirstname',  
  'UserLastname',  
  'CustomerID',  
  'CustomerName',  
  'From',  
  'Subject',  
  'AccountedTime',  
  'ArticleTree',  
  'SolutionInMin',  
  'SolutionDiffInMin',
```

```
'FirstResponseInMin',
'FirstResponseDiffInMin'
];
```

Ticket::Frontend::AgentTicketSearch###ArticleCreateTime

Includes article create times in the ticket search of the agent interface.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'ArticleCreateTime'} = '0';
```

Ticket::Frontend::AgentTicketSearch###Defaults###Fulltext

Defines the default shown ticket search attribute for ticket search screen.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Fulltext'} = '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###TicketNumber

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketNumber'} = '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###Title

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Title'} = '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###From

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'From'} = '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###To

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'To'} = '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###Cc

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Cc'} = '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###Subject

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Subject'} = '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###Body

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'Body'} = '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###CustomerID

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'CustomerID'} = '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###CustomerUserLogin

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'CustomerUserLogin'} = '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###StateIDs

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'StateIDs'} = [];
```

Ticket::Frontend::AgentTicketSearch###Defaults###QueueIDs

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'QueueIDs'} = [];
```

Ticket::Frontend::AgentTicketSearch###Defaults###ServiceIDs

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'ServiceIDs'} = [];
```

Ticket::Frontend::AgentTicketSearch###Defaults###SLAIDs

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'SLAIDs'} = [];
```

Ticket::Frontend::AgentTicketSearch###Defaults###TicketCreateTimePoint

Default data to use on attribute for ticket search screen. Example:
"TicketCreateTimePointFormat=year;TicketCreateTimePointStart=Last;TicketCreateTimePoint=2;"

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCreateTimePoint'}  
= '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###TicketCreateTimeSlot

Default data to use on attribute for ticket search screen. Example:
"TicketCreateTimeStartYear=2010;TicketCreateTimeStartMonth=10;TicketCreateTimeStartDay=4;T

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCreateTimeSlot'}  
= '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###TicketChangeTimePoint

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketChangeTimePoint'}  
= '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###TicketChangeTimeSlot

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketChangeTimeSlot'}  
= '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###TicketCloseTimePoint

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCloseTimePoint'}  
= '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###TicketCloseTimeSlot

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketCloseTimeSlot'} =  
'';
```

Ticket::Frontend::AgentTicketSearch###Defaults###TicketEscalationTimePoint

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketEscalationTimePoint'} = '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###TicketEscalationTimeSlot

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'TicketEscalationTimeSlot'} = '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###ArticleCreateTimePoint

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'ArticleCreateTimePoint'} = '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###ArticleCreateTimeSlot

Defines the default shown ticket search attribute for ticket search screen.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'ArticleCreateTimeSlot'} = '';
```

Ticket::Frontend::AgentTicketSearch###Defaults###SearchInArchive

Defines the default shown ticket search attribute for ticket search screen (AllTickets/ArchivedTickets/NotArchivedTickets).

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'SearchInArchive'} = '';
```

Ticket::Frontend::AgentTicketSearch###DynamicField

Dynamic fields shown in the ticket search screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and shown by default.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'DynamicField'} = {};
```

Ticket::Frontend::AgentTicketSearch###Defaults###DynamicField

Defines the default shown ticket search attribute for ticket search screen. Example: "Key" must have the name of the Dynamic Field in this case 'X', "Content" must have the value of the Dynamic Field depending on the Dynamic Field type, Text: 'a text', Dropdown: '1', Date/Time: 'Search_DynamicField_XTimeSlotStartYear=1974; Search_DynamicField_XTimeSlotStartMonth=01;

```

Search_DynamicField_XTimeSlotStartDay=26;
Search_DynamicField_XTimeSlotStartHour=00;
Search_DynamicField_XTimeSlotStartMinute=00;
Search_DynamicField_XTimeSlotStartSecond=00;
Search_DynamicField_XTimeSlotStopYear=2013;
Search_DynamicField_XTimeSlotStopMonth=01;
Search_DynamicField_XTimeSlotStopDay=26;
Search_DynamicField_XTimeSlotStopHour=23;
Search_DynamicField_XTimeSlotStopMinute=59;
Search_DynamicField_XTimeSlotStopSecond=59;
'Search_DynamicField_XTimePointFormat=week;
Search_DynamicField_XTimePointStart=Before;
Search_DynamicField_XTimePointValue=7';

```

and or

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'Defaults'}->{'DynamicField'} = {};
```

Ticket::Frontend::AgentTicketSearch###SearchCSVDynamicField

Dynamic Fields used to export the search result in CSV format.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'SearchCSVDynamicField'} = {};
```

Ticket::Frontend::AgentTicketSearch###DefaultColumns

Columns that can be filtered in the ticket search result view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes, Dynamic Fields (DynamicField_NameX) and Customer attributes (e.g. CustomerUserPhone, CustomerCompanyName, ...) are allowed.

Standardwert:

```

$Self->{'Ticket::Frontend::AgentTicketSearch'}->{'DefaultColumns'} = {
  'Age' => '2',
  'Changed' => '1',
  'Created' => '1',
  'CustomerCompanyName' => '1',
  'CustomerID' => '2',
  'CustomerName' => '1',
  'CustomerUserID' => '1',
  'EscalationResponseTime' => '1',
  'EscalationSolutionTime' => '1',
  'EscalationTime' => '1',
  'EscalationUpdateTime' => '1',
  'Lock' => '2',
  'Owner' => '2',
  'PendingTime' => '1',
  'Priority' => '1',
  'Queue' => '2',
  'Responsible' => '1',
  'SLA' => '1',
  'Service' => '1',
  'State' => '2',
  'TicketNumber' => '2',
  'Title' => '2',
  'Type' => '1'
};

```

Ticket → Frontend::Agent::Ticket::ViewService

Ticket::Frontend::AgentTicketService###StripEmptyLines

Strips empty lines on the ticket preview in the service view.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketService'}->{'StripEmptyLines'} = '0';
```

Ticket::Frontend::AgentTicketService###ViewAllPossibleTickets

Shows all both ro and rw tickets in the service view.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketService'}->{'ViewAllPossibleTickets'} = '0';
```

Ticket::Frontend::AgentTicketService###ServiceSort

Sorts the tickets (ascendingly or descendingly) when a single queue is selected in the service view and after the tickets are sorted by priority. Values: 0 = ascending (oldest on top, default), 1 = descending (youngest on top). Use the ServiceID for the key and 0 or 1 for value.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketService'}->{'ServiceSort'} = {
  '3' => '0',
  '7' => '1'
};
```

Ticket::Frontend::AgentTicketService###SortBy::Default

Defines the default sort criteria for all services displayed in the service view.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketService'}->{'SortBy::Default'} = 'Age';
```

Ticket::Frontend::AgentTicketService###PreSort::ByPriority

Defines if a pre-sorting by priority should be done in the service view.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketService'}->{'PreSort::ByPriority'} = '1';
```

Ticket::Frontend::AgentTicketService###Order::Default

Defines the default sort order for all services in the service view, after priority sort.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketService'}->{'Order::Default'} = 'Up';
```

Ticket::Frontend::AgentTicketService###DefaultColumns

Columns that can be filtered in the service view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes, Dynamic Fields (DynamicField_NameX) and Customer attributes (e.g. CustomerUserName, CustomerCompanyname, ...) are allowed.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketService'}->{'DefaultColumns'} = {
  'Age' => '2',
  'Changed' => '1',
};
```



```
'Created' => '1',
'CustomerCompanyName' => '1',
'CustomerID' => '2',
'CustomerName' => '1',
'CustomerUserID' => '1',
'EscalationResponseTime' => '1',
'EscalationSolutionTime' => '1',
'EscalationTime' => '1',
'EscalationUpdateTime' => '1',
'Lock' => '2',
'Owner' => '2',
'PendingTime' => '1',
'Priority' => '1',
'Queue' => '2',
'Responsible' => '1',
'SLA' => '1',
'Service' => '2',
'State' => '2',
'TicketNumber' => '2',
'Title' => '2',
'Type' => '1'
};
```

Ticket → Frontend::Agent::Ticket::ViewStatus

Ticket::Frontend::AgentTicketStatusView###ViewableTicketsPage

Shows all open tickets (even if they are locked) in the status view of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'ViewableTicketsPage'} = '50';
```

Ticket::Frontend::AgentTicketStatusView###SortBy::Default

Defines the default ticket attribute for ticket sorting in the status view of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'SortBy::Default'} = 'Age';
```

Ticket::Frontend::AgentTicketStatusView###Order::Default

Defines the default ticket order (after priority sort) in the status view of the agent interface. Up: oldest on top. Down: latest on top.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'Order::Default'} = 'Down';
```

Ticket::Frontend::AgentTicketStatusView###DefaultColumns

Columns that can be filtered in the status view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes, Dynamic Fields (DynamicField_NameX) and Customer attributes (e.g. CustomerUserPhone, CustomerCompanyName, ...) are allowed.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketStatusView'}->{'DefaultColumns'} = {
  'Age' => '2',
  'Changed' => '1',
  'Created' => '1',
```

```
'CustomerCompanyName' => '1',
'CustomerID' => '2',
'CustomerName' => '1',
'CustomerUserID' => '1',
'EscalationResponseTime' => '1',
'EscalationSolutionTime' => '1',
'EscalationTime' => '1',
'EscalationUpdateTime' => '1',
'Lock' => '2',
'Owner' => '2',
'PendingTime' => '1',
'Priority' => '1',
'Queue' => '2',
'Responsible' => '1',
'SLA' => '1',
'Service' => '1',
'State' => '2',
'TicketNumber' => '2',
'Title' => '2',
'Type' => '1'
};
```

Ticket → Frontend::Agent::Ticket::ViewWatch

Ticket::Frontend::AgentTicketWatchView###SortBy::Default

Defines the default ticket attribute for ticket sorting in the watch view of the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketWatchView'}->{'SortBy::Default'} = 'Age';
```

Ticket::Frontend::AgentTicketWatchView###Order::Default

Defines the default ticket order in the watch view of the agent interface. Up: oldest on top. Down: latest on top.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketWatchView'}->{'Order::Default'} = 'Up';
```

Ticket::Frontend::AgentTicketWatchView###DefaultColumns

Columns that can be filtered in the watch view of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default. Note: Only Ticket attributes, Dynamic Fields (DynamicField_NameX) and Customer attributes (e.g. CustomerUserID, CustomerCompanyName, ...) are allowed.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketWatchView'}->{'DefaultColumns'} = {
  'Age' => '2',
  'Changed' => '1',
  'Created' => '1',
  'CustomerCompanyName' => '1',
  'CustomerID' => '2',
  'CustomerName' => '1',
  'CustomerUserID' => '1',
  'EscalationResponseTime' => '1',
  'EscalationSolutionTime' => '1',
  'EscalationTime' => '1',
  'EscalationUpdateTime' => '1',
  'Lock' => '2',
  'Owner' => '2',
  'PendingTime' => '1',
  'Priority' => '1',
```

```
'Queue' => '2',
'Responsible' => '1',
'SLA' => '1',
'Service' => '1',
'State' => '2',
'TicketNumber' => '2',
'Title' => '2',
'Type' => '1'
};
```

Ticket → Frontend::Agent::Ticket::ViewZoom

Ticket::Frontend::PlainView

Shows a link to see a zoomed email ticket in plain text.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::PlainView'} = '0';
```

Ticket::Frontend::ZoomExpand

Shows all the articles of the ticket (expanded) in the zoom view.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ZoomExpand'} = '0';
```

Ticket::Frontend::ZoomExpandSort

Shows the articles sorted normally or in reverse, under ticket zoom in the agent interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ZoomExpandSort'} = 'reverse';
```

Ticket::ZoomAttachmentDisplayCount

Shows a count of icons in the ticket zoom, if the article has attachments.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::ZoomAttachmentDisplayCount'} = '20';
```

Ticket::ZoomTimeDisplay

Displays the accounted time for an article in the ticket zoom view.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::ZoomTimeDisplay'} = '0';
```

Ticket::UseArticleColors

Shows colors for different article types in the article table.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::UseArticleColors'} = '1';
```

Ticket::Frontend::TicketArticleFilter

Activates the article filter in the zoom view to specify which articles should be shown.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::TicketArticleFilter'} = '0';
```

Ticket::Frontend::HTMLArticleHeightDefault

Set the default height (in pixels) of inline HTML articles in AgentTicketZoom.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::HTMLArticleHeightDefault'} = '100';
```

Ticket::Frontend::HTMLArticleHeightMax

Set the maximum height (in pixels) of inline HTML articles in AgentTicketZoom.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::HTMLArticleHeightMax'} = '2500';
```

Ticket::Frontend::MaxArticlesZoomExpand

The maximal number of articles expanded on a single page in AgentTicketZoom.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::MaxArticlesZoomExpand'} = '400';
```

Ticket::Frontend::MaxArticlesPerPage

The maximal number of articles shown on a single page in AgentTicketZoom.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::MaxArticlesPerPage'} = '1000';
```

Ticket::Frontend::ZoomRichTextForce

Show article as rich text even if rich text writing is disabled.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::ZoomRichTextForce'} = '1';
```

Ticket::Frontend::AgentTicketZoom###DynamicField

Dynamic fields shown in the sidebar of the ticket zoom screen of the agent interface.
Possible settings: 0 = Disabled, 1 = Enabled.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketZoom'}->{'DynamicField'} = {};
```

Ticket::Frontend::AgentTicketZoom###Widgets###0100-TicketInformation

AgentTicketZoom widget that displays ticket data in the side bar.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketZoom'}->{'Widgets'}->{'0100-TicketInformation'} =
{
  'Location' => 'Sidebar',
  'Module' => 'Kernel::Output::HTML::TicketZoom::TicketInformation'
};
```

Ticket::Frontend::AgentTicketZoom###Widgets###0200-CustomerInformation

AgentTicketZoom widget that displays customer information for the ticket in the side bar.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketZoom'}->{'Widgets'}->{'0200-CustomerInformation'}
= {
  'Location' => 'Sidebar',
  'Module' => 'Kernel::Output::HTML::TicketZoom::CustomerInformation'
};
```

Ticket::Frontend::AgentTicketZoom###Widgets###0300-LinkTable

AgentTicketZoom widget that displays a table of objects linked to the ticket.

Standardwert:

```
$Self->{'Ticket::Frontend::AgentTicketZoom'}->{'Widgets'}->{'0300-LinkTable'} = {
  'Module' => 'Kernel::Output::HTML::TicketZoom::LinkTable'
};
```

Ticket::Frontend::ZoomCollectMeta

Whether or not to collect meta information from articles using filters configured in Ticket::Frontend::ZoomCollectMetaFilters.

Standardwert:

```
$Self->{'Ticket::Frontend::ZoomCollectMeta'} = '0';
```

Ticket::Frontend::ZoomCollectMetaFilters###CVE-Mitre

Defines a filter to collect CVE numbers from article texts in AgentTicketZoom. The results will be displayed in a meta box next to the article. Fill in URLPreview if you would like to see a preview when moving your mouse cursor above the link element. This could be the same URL as in URL, but also an alternate one. Please note that some websites deny being displayed within an iframe (e.g. Google) and thus won't work with the preview mode.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::ZoomCollectMetaFilters'}->{'CVE-Mitre'} = {
  'Meta' => {
    'Name' => 'CVE Mitre',
    'Target' => '_blank',
    'URL' => 'http://cve.mitre.org/cgi-bin/cvename.cgi?name=<MATCH1>-<MATCH2>-<MATCH3>',
    'URLPreview' => 'http://cve.mitre.org/cgi-bin/cvename.cgi?name=<MATCH1>-<MATCH2>-<MATCH3>'
  },
  'RegExp' => [
    '(CVE|CAN)\-(\d{3,4})\-(\d{2,})'
  ]
};
```

Ticket::Frontend::ZoomCollectMetaFilters###CVE-Google

Defines a filter to collect CVE numbers from article texts in AgentTicketZoom. The results will be displayed in a meta box next to the article. Fill in URLPreview if you

would like to see a preview when moving your mouse cursor above the link element. This could be the same URL as in URL, but also an alternate one. Please note that some websites deny being displayed within an iframe (e.g. Google) and thus won't work with the preview mode.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::ZoomCollectMetaFilters'}->{'CVE-Google'} = {
  'Meta' => {
    'Name' => 'CVE Google Search',
    'Target' => '_blank',
    'URL' => 'http://google.com/search?q=<MATCH1>-<MATCH2>-<MATCH3>',
    'URLPreview' => ''
  },
  'RegExp' => [
    '(CVE|CAN)\\-(\\d{3,4})\\-(\\d{2,})'
  ]
};
```

Ticket → Frontend::Agent::TicketOverview

Ticket::Frontend::Overview###Small

Allows having a small format ticket overview (CustomerInfo => 1 - shows also the customer information).

Standardwert:

```
$Self->{'Ticket::Frontend::Overview'}->{'Small'} = {
  'CustomerInfo' => '1',
  'Module' => 'Kernel::Output::HTML::TicketOverview::Small',
  'ModulePriority' => '100',
  'Name' => 'Small',
  'NameShort' => 'S'
};
```

Ticket::Frontend::OverviewSmall###ColumnHeader

Shows either the last customer article's subject or the ticket title in the small format overview.

Standardwert:

```
$Self->{'Ticket::Frontend::OverviewSmall'}->{'ColumnHeader'} = 'LastCustomerSubject';
```

Ticket::Frontend::Overview###Medium

Allows having a medium format ticket overview (CustomerInfo => 1 - shows also the customer information).

Standardwert:

```
$Self->{'Ticket::Frontend::Overview'}->{'Medium'} = {
  'CustomerInfo' => '0',
  'Module' => 'Kernel::Output::HTML::TicketOverview::Medium',
  'ModulePriority' => '200',
  'Name' => 'Medium',
  'NameShort' => 'M',
  'OverviewMenuModules' => '1',
  'TicketActionsPerTicket' => '1'
};
```

Ticket::Frontend::Overview###Preview

Shows a preview of the ticket overview (CustomerInfo => 1 - shows also Customer-Info, CustomerInfoMaxSize max. size in characters of Customer-Info).

Standardwert:

```
$Self->{'Ticket::Frontend::Overview'}->{'Preview'} = {
  'CustomerInfo' => '0',
  'CustomerInfoMaxSize' => '18',
  'DefaultPreViewLines' => '25',
  'DefaultViewNewLine' => '90',
  'Module' => 'Kernel::Output::HTML::TicketOverview::Preview',
  'ModulePriority' => '300',
  'Name' => 'Large',
  'NameShort' => 'L',
  'OverviewMenuModules' => '1',
  'StripEmptyLines' => '0',
  'TicketActionsPerTicket' => '1'
};
```

Ticket::Frontend::Overview::PreviewArticleSenderTypes

Defines which article sender types should be shown in the preview of a ticket.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::Overview::PreviewArticleSenderTypes'} = {
  'agent' => '1',
  'customer' => '1',
  'system' => '1'
};
```

Ticket::Frontend::Overview::PreviewArticleLimit

Sets the count of articles visible in preview mode of ticket overviews.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::Overview::PreviewArticleLimit'} = '5';
```

Ticket::Frontend::Overview::PreviewArticleTypeExpanded

Defines which article type should be expanded when entering the overview. If nothing defined, latest article will be expanded.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::Overview::PreviewArticleTypeExpanded'} = '';
```

Ticket::Frontend::OverviewSmall###DynamicField

Dynamic fields shown in the ticket small format overview screen of the agent interface. Possible settings: 0 = Disabled, 1 = Available, 2 = Enabled by default.

Standardwert:

```
$Self->{'Ticket::Frontend::OverviewSmall'}->{'DynamicField'} = {};
```

Ticket::Frontend::OverviewMedium###DynamicField

Dynamic fields shown in the ticket medium format overview screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.

Standardwert:

```
$Self->{'Ticket::Frontend::OverviewMedium'}->{'DynamicField'} = {};
```

Ticket::Frontend::OverviewPreview###DynamicField

Dynamic fields shown in the ticket preview format overview screen of the agent interface. Possible settings: 0 = Disabled, 1 = Enabled.

Standardwert:

```
$Self->{'Ticket::Frontend::OverviewPreview'}->{'DynamicField'} = {};
```

Ticket → Frontend::Agent::ToolBarModule

Frontend::ToolBarModule###110-Ticket::AgentTicketQueue

Toolbar Item for a shortcut. 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".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::ToolBarModule'}->{'110-Ticket::AgentTicketQueue'} = {
  'AccessKey' => 'q',
  'Action' => 'AgentTicketQueue',
  'CssClass' => 'QueueView',
  'Icon' => 'fa fa-folder',
  'Link' => 'Action=AgentTicketQueue',
  'Module' => 'Kernel::Output::HTML::ToolBar::Link',
  'Name' => 'Queue view',
  'Priority' => '1010010'
};
```

Frontend::ToolBarModule###120-Ticket::AgentTicketStatus

Toolbar Item for a shortcut. 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".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::ToolBarModule'}->{'120-Ticket::AgentTicketStatus'} = {
  'AccessKey' => 'S',
  'Action' => 'AgentTicketStatusView',
  'CssClass' => 'StatusView',
  'Icon' => 'fa fa-list-ol',
  'Link' => 'Action=AgentTicketStatusView',
  'Module' => 'Kernel::Output::HTML::ToolBar::Link',
  'Name' => 'Status view',
  'Priority' => '1010020'
};
```

Frontend::ToolBarModule###130-Ticket::AgentTicketEscalation

Toolbar Item for a shortcut. 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".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::ToolBarModule'}->{'130-Ticket::AgentTicketEscalation'} = {
  'AccessKey' => 'w',
  'Action' => 'AgentTicketEscalationView',
  'CssClass' => 'EscalationView',
  'Icon' => 'fa fa-exclamation',
  'Link' => 'Action=AgentTicketEscalationView',
  'Module' => 'Kernel::Output::HTML::ToolBar::Link',
  'Name' => 'Escalation view',
  'Priority' => '1010030'
};
```

Frontend::ToolBarModule###140-Ticket::AgentTicketPhone

Toolbar Item for a shortcut. 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".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::ToolBarModule'}->{'140-Ticket::AgentTicketPhone'} = {
  'AccessKey' => '',
  'Action' => 'AgentTicketPhone',
  'CssClass' => 'PhoneTicket',
  'Icon' => 'fa fa-phone',
  'Link' => 'Action=AgentTicketPhone',
  'Module' => 'Kernel::Output::HTML::ToolBar::Link',
  'Name' => 'New phone ticket',
  'Priority' => '1020010'
};
```

Frontend::ToolBarModule###150-Ticket::AgentTicketEmail

Toolbar Item for a shortcut. 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".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::ToolBarModule'}->{'150-Ticket::AgentTicketEmail'} = {
  'AccessKey' => '',
  'Action' => 'AgentTicketEmail',
  'CssClass' => 'EmailTicket',
  'Icon' => 'fa fa-envelope',
  'Link' => 'Action=AgentTicketEmail',
  'Module' => 'Kernel::Output::HTML::ToolBar::Link',
  'Name' => 'New email ticket',
  'Priority' => '1020020'
};
```

Frontend::ToolBarModule###160-Ticket::AgentTicketProcess

Toolbar Item for a shortcut. 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".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::ToolBarModule'}->{'160-Ticket::AgentTicketProcess'} = {
  'AccessKey' => '',
  'Action' => 'AgentTicketProcess',
  'CssClass' => 'ProcessTicket',
  'Icon' => 'fa fa-th-large',
  'Link' => 'Action=AgentTicketProcess',
  'Module' => 'Kernel::Output::HTML::ToolBar::Link',
  'Name' => 'New process ticket',
  'Priority' => '1020030'
};
```

Frontend::ToolBarModule###170-Ticket::TicketResponsible

Agent interface notification module to see the number of tickets an agent is responsible for. 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".

Standardwert:

```
$Self->{'Frontend::ToolBarModule'}->{'170-Ticket::TicketResponsible'} = {
  'AccessKey' => 'r',
  'AccessKeyNew' => '',
  'AccessKeyReached' => '',
  'CssClass' => 'Responsible',
  'CssClassNew' => 'Responsible New',
  'CssClassReached' => 'Responsible Reached',
  'Icon' => 'fa fa-user',
  'IconNew' => 'fa fa-user',
  'IconReached' => 'fa fa-user',
  'Module' => 'Kernel::Output::HTML::ToolBar::TicketResponsible',
};
```

```
'Priority' => '1030010'
};
```

Frontend::ToolBarModule###180-Ticket::TicketWatcher

Agent interface notification module to see the number of watched tickets. 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".

Standardwert:

```
$Self->{'Frontend::ToolBarModule'}->{'180-Ticket::TicketWatcher'} = {
  'AccessKey' => '',
  'AccessKeyNew' => '',
  'AccessKeyReached' => '',
  'CssClass' => 'Watcher',
  'CssClassNew' => 'Watcher New',
  'CssClassReached' => 'Watcher Reached',
  'Icon' => 'fa fa-eye',
  'IconNew' => 'fa fa-eye',
  'IconReached' => 'fa fa-eye',
  'Module' => 'Kernel::Output::HTML::ToolBar::TicketWatcher',
  'Priority' => '1030020'
};
```

Frontend::ToolBarModule###190-Ticket::TicketLocked

Agent interface notification module to see the number of locked tickets. 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".

Standardwert:

```
$Self->{'Frontend::ToolBarModule'}->{'190-Ticket::TicketLocked'} = {
  'AccessKey' => 'k',
  'AccessKeyNew' => '',
  'AccessKeyReached' => '',
  'CssClass' => 'Locked',
  'CssClassNew' => 'Locked New',
  'CssClassReached' => 'Locked Reached',
  'Icon' => 'fa fa-lock',
  'IconNew' => 'fa fa-lock',
  'IconReached' => 'fa fa-lock',
  'Module' => 'Kernel::Output::HTML::ToolBar::TicketLocked',
  'Priority' => '1030030'
};
```

Frontend::ToolBarModule###200-Ticket::AgentTicketService

Agent interface notification module to see the number of tickets in My Services. 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".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::ToolBarModule'}->{'200-Ticket::AgentTicketService'} = {
  'CssClass' => 'ServiceView',
  'Icon' => 'fa fa-wrench',
  'Module' => 'Kernel::Output::HTML::ToolBar::TicketService',
  'Priority' => '1030035'
};
```

Frontend::ToolBarModule###210-Ticket::TicketSearchProfile

Agent interface module to access search profiles via nav bar. 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".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::ToolBarModule'}->{'210-Ticket::TicketSearchProfile'} = {
  'Block' => 'ToolBarSearchProfile',
  'Description' => 'Search template',
  'MaxWidth' => '40',
  'Module' => 'Kernel::Output::HTML::ToolBar::TicketSearchProfile',
  'Name' => 'Search template',
  'Priority' => '1990010'
};
```

Frontend::ToolBarModule###220-Ticket::TicketSearchFulltext

Agent interface module to access fulltext search via nav bar. 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".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::ToolBarModule'}->{'220-Ticket::TicketSearchFulltext'} = {
  'Block' => 'ToolBarSearchFulltext',
  'CSS' => 'Core.Agent.Toolbar.FulltextSearch.css',
  'Description' => 'Fulltext search',
  'Module' => 'Kernel::Output::HTML::ToolBar::Generic',
  'Name' => 'Fulltext search',
  'Priority' => '1990020',
  'Size' => '10'
};
```

Frontend::ToolBarModule###230-CICSearchCustomerID

Agent interface module to access CIC search via nav bar. 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".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::ToolBarModule'}->{'230-CICSearchCustomerID'} = {
  'Block' => 'ToolBarCICSearchCustomerID',
  'CSS' => 'Core.Agent.Toolbar.CICSearch.css',
  'Description' => 'CustomerID search',
  'Module' => 'Kernel::Output::HTML::ToolBar::Generic',
  'Name' => 'CustomerID search',
  'Priority' => '1990030',
  'Size' => '10'
};
```

Frontend::ToolBarModule###240-CICSearchCustomerUser

Agent interface module to access CIC search via nav bar. 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".

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::ToolBarModule'}->{'240-CICSearchCustomerUser'} = {
  'Block' => 'ToolBarCICSearchCustomerUser',
  'CSS' => 'Core.Agent.Toolbar.CICSearch.css',
  'Description' => 'Customer user search',
  'Module' => 'Kernel::Output::HTML::ToolBar::Generic',
  'Name' => 'Customer user search',
  'Priority' => '1990040',
  'Size' => '10'
};
```

Ticket → Frontend::Customer

Ticket::Frontend::CustomerDisableCompanyTicketAccess

This option will deny the access to customer company tickets, which are not created by the customer user.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerDisableCompanyTicketAccess'} = '0';
```

Ticket::Frontend::CustomerTicketOverviewCustomEmptyText

Custom text for the page shown to customers that have no tickets yet (if you need those text translated add them to a custom translation module).

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketOverviewCustomEmptyText'} = {
  'Button' => 'Create your first ticket',
  'Text' => 'Please click the button below to create your first ticket.',
  'Title' => 'Welcome!'
};
```

Frontend::CustomerUser::Item###15-OpenTickets

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.

Standardwert:

```
$Self->{'Frontend::CustomerUser::Item'}->{'15-OpenTickets'} = {
  'Action' => 'AgentTicketSearch',
  'Attributes' => 'StateType=Open;',
  'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css',
  'CSSClassNoOpenTicket' => 'NoOpenTicket',
  'CSSClassOpenTicket' => 'OpenTicket',
  'CustomerUserLogin' => '0',
  'IconNameNoOpenTicket' => 'fa-check-circle',
  'IconNameOpenTicket' => 'fa-exclamation-circle',
  'Module' => 'Kernel::Output::HTML::CustomerUser::GenericTicket',
  'Subaction' => 'Search',
  'Target' => '_blank',
  'Text' => 'Open tickets (customer)'
};
```

Frontend::CustomerUser::Item###16-OpenTicketsForCustomerUserLogin

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::CustomerUser::Item'}->{'16-OpenTicketsForCustomerUserLogin'} = {
  'Action' => 'AgentTicketSearch',
  'Attributes' => 'StateType=Open;',
  'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css',
  'CSSClassNoOpenTicket' => 'NoOpenTicket',
  'CSSClassOpenTicket' => 'OpenTicket',
  'CustomerUserLogin' => '1',
  'IconNameNoOpenTicket' => 'fa-check-circle',
  'IconNameOpenTicket' => 'fa-exclamation-circle',
  'Module' => 'Kernel::Output::HTML::CustomerUser::GenericTicket',
};
```

```
'Subaction' => 'Search',
'Target' => '_blank',
'Text' => 'Open tickets (customer user)'
};
```

Frontend::CustomerUser::Item###17-ClosedTickets

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::CustomerUser::Item'}->{'17-ClosedTickets'} = {
  'Action' => 'AgentTicketSearch',
  'Attributes' => 'StateType=Closed;',
  'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css',
  'CSSClassNoOpenTicket' => 'NoOpenTicket',
  'CSSClassOpenTicket' => 'OpenTicket',
  'CustomerUserLogin' => '0',
  'IconNameNoOpenTicket' => 'fa-power-off',
  'IconNameOpenTicket' => 'fa-power-off',
  'Module' => 'Kernel::Output::HTML::CustomerUser::GenericTicket',
  'Subaction' => 'Search',
  'Target' => '_blank',
  'Text' => 'Closed tickets (customer)'
};
```

Frontend::CustomerUser::Item###18-ClosedTicketsForCustomerUserLogin

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.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Frontend::CustomerUser::Item'}->{'18-ClosedTicketsForCustomerUserLogin'} = {
  'Action' => 'AgentTicketSearch',
  'Attributes' => 'StateType=Closed;',
  'CSS' => 'Core.Agent.CustomerUser.OpenTicket.css',
  'CSSClassNoOpenTicket' => 'NoOpenTicket',
  'CSSClassOpenTicket' => 'OpenTicket',
  'CustomerUserLogin' => '1',
  'IconNameNoOpenTicket' => 'fa-power-off',
  'IconNameOpenTicket' => 'fa-power-off',
  'Module' => 'Kernel::Output::HTML::CustomerUser::GenericTicket',
  'Subaction' => 'Search',
  'Target' => '_blank',
  'Text' => 'Closed tickets (customer user)'
};
```

CustomerFrontend::CommonParam###Action

Defines the default used Frontend-Module if no Action parameter given in the url on the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerFrontend::CommonParam'}->{'Action'} = 'CustomerTicketOverview';
```

CustomerFrontend::CommonParam###TicketID

Default ticket ID used by the system in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerFrontend::CommonParam'}->{'TicketID'} = '';
```

Ticket → Frontend::Customer::ModuleMetaHead

CustomerFrontend::HeaderMetaModule###2-TicketSearch

Module to generate html OpenSearch profile for short ticket search in the customer interface.

Standardwert:

```
$Self->{'CustomerFrontend::HeaderMetaModule'}->{'2-TicketSearch'} = {
  'Action' => 'CustomerTicketSearch',
  'Module' => 'Kernel::Output::HTML::HeaderMeta::CustomerTicketSearch'
};
```

Ticket → Frontend::Customer::ModuleRegistration

CustomerFrontend::Module###CustomerTicketOverview

Frontend module registration for the customer interface.

Standardwert:

```
$Self->{'CustomerFrontend::Module'}->{'CustomerTicketOverview'} = {
  'Description' => 'Overview of customer tickets.',
  'NavBar' => [
    {
      'AccessKey' => 'm',
      'Block' => '',
      'Description' => 'Tickets.',
      'Link' => 'Action=CustomerTicketOverview;Subaction=MyTickets',
      'LinkOption' => '',
      'Name' => 'Tickets',
      'NavBar' => 'Ticket',
      'Prio' => '100',
      'Type' => 'Menu'
    },
    {
      'AccessKey' => '',
      'Block' => '',
      'Description' => 'My Tickets.',
      'Link' => 'Action=CustomerTicketOverview;Subaction=MyTickets',
      'LinkOption' => '',
      'Name' => 'My Tickets',
      'NavBar' => 'Ticket',
      'Prio' => '110',
      'Type' => 'Submenu'
    },
    {
      'AccessKey' => 'M',
      'Block' => '',
      'Description' => 'Company Tickets.',
      'Link' => 'Action=CustomerTicketOverview;Subaction=CompanyTickets',
      'LinkOption' => '',
      'Name' => 'Company Tickets',
      'NavBar' => 'Ticket',
      'Prio' => '120',
      'Type' => 'Submenu'
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'Overview'
};
```

CustomerFrontend::Module###CustomerTicketMessage

Frontend module registration for the customer interface.

Standardwert:

```

$self->{'CustomerFrontend::Module'}->{'CustomerTicketMessage'} = {
  'Description' => 'Create tickets.',
  'Loader' => {
    'JavaScript' => [
      'Core.Customer.TicketMessage.js'
    ]
  },
  'NavBar' => [
    {
      'AccessKey' => 'n',
      'Block' => '',
      'Description' => 'Create new Ticket.',
      'Link' => 'Action=CustomerTicketMessage',
      'LinkOption' => '',
      'Name' => 'New Ticket',
      'NavBar' => 'Ticket',
      'Prio' => '100',
      'Type' => 'Submenu'
    }
  ],
  'NavBarName' => 'Ticket',
  'Title' => 'New Ticket'
};

```

CustomerFrontend::Module###CustomerTicketZoom

Frontend module registration for the customer interface.

Standardwert:

```

$self->{'CustomerFrontend::Module'}->{'CustomerTicketZoom'} = {
  'Description' => 'Ticket zoom view.',
  'Loader' => {
    'JavaScript' => [
      'Core.Customer.TicketZoom.js',
      'Core.UI.Popup.js'
    ]
  },
  'NavBarName' => 'Ticket',
  'Title' => 'Zoom'
};

```

CustomerFrontend::Module###CustomerTicketPrint

Frontend module registration for the customer interface.

Standardwert:

```

$self->{'CustomerFrontend::Module'}->{'CustomerTicketPrint'} = {
  'Description' => 'Customer Ticket Print Module.',
  'NavBarName' => '',
  'Title' => 'Print'
};

```

CustomerFrontend::Module###CustomerTicketAttachment

Frontend module registration for the customer interface.

Standardwert:

```

$self->{'CustomerFrontend::Module'}->{'CustomerTicketAttachment'} = {
  'Description' => 'To download attachments.',
  'NavBarName' => '',
  'Title' => ''
};

```

CustomerFrontend::Module###CustomerTicketSearch

Frontend module registration for the customer interface.

Standardwert:

```

$self->{'CustomerFrontend::Module'}->{'CustomerTicketSearch'} = {

```

```
'Description' => 'Customer ticket search.',
'Loader' => {
  'JavaScript' => [
    'Core.Customer.TicketSearch.js'
  ]
},
'NavBar' => [
  {
    'AccessKey' => 's',
    'Block' => '',
    'Description' => 'Search.',
    'Link' => 'Action=CustomerTicketSearch',
    'LinkOption' => '',
    'Name' => 'Search',
    'NavBar' => 'Ticket',
    'Prio' => '300',
    'Type' => 'Submenu'
  }
],
'NavBarName' => 'Ticket',
'Title' => 'Search'
};
```

Ticket → Frontend::Customer::Preferences

CustomerPreferencesGroups###ShownTickets

Defines all the parameters for the ShownTickets object in the customer preferences of the customer interface.

Standardwert:

```
$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::Preferences::Generic',
  'PrefKey' => 'UserShowTickets',
  'Prio' => '4000'
};
```

CustomerPreferencesGroups###RefreshTime

Defines all the parameters for the RefreshTime object in the customer preferences of the customer interface.

Standardwert:

```
$Self->{'CustomerPreferencesGroups'}->{'RefreshTime'} = {
  'Active' => '1',
  'Column' => 'User Profile',
  'Data' => {
    '0' => 'off',
    '10' => '10 minutes',
    '15' => '15 minutes',
    '2' => ' 2 minutes',
    '5' => ' 5 minutes',
    '7' => ' 7 minutes'
  },
  'DataSelected' => '0',
  'Key' => 'Refresh interval',
  'Label' => 'Ticket overview',
  'Module' => 'Kernel::Output::HTML::Preferences::Generic',
  'PrefKey' => 'UserRefreshTime',
};
```



```
'Prio' => '4000'  
};
```

Ticket → Frontend::Customer::Ticket::ViewNew

Ticket::Frontend::CustomerTicketMessage###NextScreenAfterNewTicket

Determines the next screen after new customer ticket in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'NextScreenAfterNewTicket'} =  
'CustomerTicketOverview';
```

Ticket::Frontend::CustomerTicketMessage###Priority

Allows customers to set the ticket priority in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'Priority'} = '1';
```

Ticket::Frontend::CustomerTicketMessage###PriorityDefault

Defines the default priority of new customer tickets in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'PriorityDefault'} = '3 normal';
```

Ticket::Frontend::CustomerTicketMessage###Queue

Allows customers to set the ticket queue in the customer interface. If this is set to 'No', QueueDefault should be configured.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'Queue'} = '1';
```

Ticket::Frontend::CustomerTicketMessage###QueueDefault

Defines the default queue for new customer tickets in the customer interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'QueueDefault'} = 'Postmaster';
```

Ticket::Frontend::CustomerTicketMessage###TicketType

Allows customers to set the ticket type in the customer interface. If this is set to 'No', TicketTypeDefault should be configured.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'TicketType'} = '1';
```

Ticket::Frontend::CustomerTicketMessage###TicketTypeDefault

Defines the default ticket type for new customer tickets in the customer interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'TicketTypeDefault'} =  
'Unclassified';
```

Ticket::Frontend::CustomerTicketMessage###Service

Allows customers to set the ticket service in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'Service'} = '1';
```

Ticket::Frontend::CustomerTicketMessage###SLA

Allows customers to set the ticket SLA in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'SLA'} = '1';
```

Ticket::Frontend::CustomerTicketMessage###ServiceMandatory

Sets if service must be selected by the customer.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'ServiceMandatory'} = '0';
```

Ticket::Frontend::CustomerTicketMessage###SLAMandatory

Sets if SLA must be selected by the customer.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'SLAMandatory'} = '0';
```

Ticket::Frontend::CustomerTicketMessage###StateDefault

Defines the default state of new customer tickets in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'StateDefault'} = 'new';
```

Ticket::Frontend::CustomerTicketMessage###ArticleType

Defines the default type for article in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'ArticleType'} = 'webrequest';
```

Ticket::Frontend::CustomerTicketMessage###SenderType

Sender type for new tickets from the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'SenderType'} = 'customer';
```

Ticket::Frontend::CustomerTicketMessage###HistoryType

Defines the default history type in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'HistoryType'} = 'WebRequestCustomer';
```

Ticket::Frontend::CustomerTicketMessage###HistoryComment

Comment for new history entries in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'HistoryComment'} = '';
```

CustomerPanelSelectionType

Defines the recipient target of the tickets ("Queue" shows all queues, "SystemAddress" shows only the queues which are assigned to system addresses) in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerPanelSelectionType'} = 'Queue';
```

CustomerPanelSelectionString

Determines the strings that will be shown as recipient (To:) of the ticket in the customer interface. For Queue as CustomerPanelSelectionType, "<Queue>" shows the names of the queues, and for SystemAddress, "<Realname> <<Email>>" shows the name and email of the recipient.

Standardwert:

```
$Self->{'CustomerPanelSelectionString'} = '<Queue>';
```

CustomerPanelOwnSelection

Determines which queues will be valid for ticket's recipients in the customer interface.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'CustomerPanelOwnSelection'} = {
  'Junk' => 'First Queue',
  'Misc' => 'Second Queue'
};
```

CustomerPanel::NewTicketQueueSelectionModule

Module for To-selection in new ticket screen in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'CustomerPanel::NewTicketQueueSelectionModule'} = 'Kernel::Output::HTML::CustomerNewTicket::QueueSelectionGeneric';
```

Ticket::Frontend::CustomerTicketMessage###DynamicField

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.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketMessage'}->{'DynamicField'} = {};
```

Ticket → Frontend::Customer::Ticket::ViewPrint

Ticket::Frontend::CustomerTicketPrint###DynamicField

Dynamic fields shown in the ticket print screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketPrint'}->{'DynamicField'} = {};
```

Ticket → Frontend::Customer::Ticket::ViewSearch

Ticket::CustomerTicketSearch::SearchLimit

Maximum number of tickets to be displayed in the result of a search in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::CustomerTicketSearch::SearchLimit'} = '5000';
```

Ticket::CustomerTicketSearch::SearchPageShown

Number of tickets to be displayed in each page of a search result in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::CustomerTicketSearch::SearchPageShown'} = '40';
```

Ticket::CustomerTicketSearch::SortBy::Default

Defines the default ticket attribute for ticket sorting in a ticket search of the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::CustomerTicketSearch::SortBy::Default'} = 'Age';
```

Ticket::CustomerTicketSearch::Order::Default

Defines the default ticket order of a search result in the customer interface. Up: oldest on top. Down: latest on top.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::CustomerTicketSearch::Order::Default'} = 'Down';
```

Ticket::Frontend::CustomerTicketSearch###ExtendedSearchCondition

Allows extended search conditions in ticket search of the customer interface. With this feature you can search e. g. with this kind of conditions like "(key1&&key2)" or "(key1||key2)".

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'ExtendedSearchCondition'} = '1';
```

Customer::TicketSearch::AllServices

If enabled, the customer can search for tickets in all services (regardless what services are assigned to the customer).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Customer::TicketSearch::AllServices'} = '0';
```

Ticket::Frontend::CustomerTicketSearch###SearchArticleCSVTree

Exports the whole article tree in search result (it can affect the system performance).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchArticleCSVTree'} = '0';
```

Ticket::Frontend::CustomerTicketSearch###SearchCSVData

Data used to export the search result in CSV format.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchCSVData'} = [
    'TicketNumber',
    'Age',
    'Created',
    'Closed',
    'State',
    'Priority',
    'Lock',
    'CustomerID',
    'CustomerName',
    'From',
    'Subject'
];
```

Ticket::Frontend::CustomerTicketSearch###DynamicField

Dynamic fields shown in the ticket search screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'DynamicField'} = {};
```

Ticket::Frontend::CustomerTicketSearch###SearchOverviewDynamicField

Dynamic fields shown in the ticket search overview results screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchOverviewDynamicField'} = {};
```

Ticket::Frontend::CustomerTicketSearch###SearchCSVDynamicField

Dynamic Fields used to export the search result in CSV format.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketSearch'}->{'SearchCSVDynamicField'} = {};
```

Ticket → Frontend::Customer::Ticket::ViewZoom

Ticket::Frontend::CustomerTicketZoom###NextScreenAfterFollowUp

Determines the next screen after the follow-up screen of a zoomed ticket in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'NextScreenAfterFollowUp'} = 'CustomerTicketOverview';
```

Ticket::Frontend::CustomerTicketZoom###ArticleType

Defines the default type of the note in the ticket zoom screen of the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'ArticleType'} = 'webrequest';
```

Ticket::Frontend::CustomerTicketZoom###SenderType

Defines the default sender type for tickets in the ticket zoom screen of the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'SenderType'} = 'customer';
```

Ticket::Frontend::CustomerTicketZoom###HistoryType

Defines the history type for the ticket zoom action, which gets used for ticket history in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'HistoryType'} = 'FollowUp';
```

Ticket::Frontend::CustomerTicketZoom###HistoryComment

Defines the history comment for the ticket zoom action, which gets used for ticket history in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'HistoryComment'} = '';
```

Ticket::Frontend::CustomerTicketZoom###Priority

Allows customers to change the ticket priority in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'Priority'} = '1';
```

Ticket::Frontend::CustomerTicketZoom###PriorityDefault

Defines the default priority of follow-up customer tickets in the ticket zoom screen in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'PriorityDefault'} = '3 normal';
```

Ticket::Frontend::CustomerTicketZoom###State

Allows choosing the next compose state for customer tickets in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'State'} = '1';
```

Ticket::Frontend::CustomerTicketZoom###StateDefault

Defines the default next state for a ticket after customer follow-up in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'StateDefault'} = 'open';
```

Ticket::Frontend::CustomerTicketZoom###StateType

Defines the next possible states for customer tickets in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'StateType'} = [  
  'open',  
  'closed'  
];
```

Ticket::Frontend::CustomerTicketZoom###AttributesView

Shows the activated ticket attributes in the customer interface (0 = Disabled and 1 = Enabled).

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'AttributesView'} = {  
  'Owner' => '0',  
  'Priority' => '1',  
  'Queue' => '1',  
  'Responsible' => '0',  
  'SLA' => '0',  
  'Service' => '0',  
  'State' => '1',  
  'Type' => '0'  
};
```

Ticket::Frontend::CustomerTicketZoom###DynamicField

Dynamic fields shown in the ticket zoom screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'DynamicField'} = {};
```

Ticket::Frontend::CustomerTicketZoom###FollowUpDynamicField

Dynamic fields options shown in the ticket reply section in the ticket zoom screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketZoom'}->{'FollowUpDynamicField'} = {};
```

Ticket → Frontend::Customer::TicketOverview

Ticket::Frontend::CustomerTicketOverviewSortable

Controls if customers have the ability to sort their tickets.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketOverviewSortable'} = '';
```

Ticket::Frontend::CustomerTicketOverview###ColumnHeader

Shows either the last customer article's subject or the ticket title in the small format overview.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketOverview'}->{'ColumnHeader'} = 'TicketTitle';
```

Ticket::Frontend::CustomerTicketOverview###Owner

Show the current owner in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketOverview'}->{'Owner'} = '0';
```

Ticket::Frontend::CustomerTicketOverview###Queue

Show the current queue in the customer interface.

Diese Einstellung kann nicht deaktiviert werden.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketOverview'}->{'Queue'} = '0';
```

Ticket::Frontend::CustomerTicketOverview###DynamicField

Dynamic fields shown in the ticket overview screen of the customer interface. Possible settings: 0 = Disabled, 1 = Enabled, 2 = Enabled and required.

Standardwert:

```
$Self->{'Ticket::Frontend::CustomerTicketOverview'}->{'DynamicField'} = {};
```

Ticket → Frontend::Queue::Preferences

QueuePreferences###Comment2

Parameters of the example queue attribute Comment2.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'QueuePreferences'}->{'Comment2'} = {
```



```
'Block' => 'TextArea',  
'Cols' => '50',  
'Desc' => 'Define the queue comment 2.',  
'Label' => 'Comment2',  
'Module' => 'Kernel::Output::HTML::QueuePreferences::Generic',  
'PrefKey' => 'Comment2',  
'Rows' => '5'  
};
```

Ticket → Frontend::SLA::Preferences

SLAPreferences###Comment2

Parameters of the example SLA attribute Comment2.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'SLAPreferences'}->{'Comment2'} = {  
'Block' => 'TextArea',  
'Cols' => '50',  
'Desc' => 'Define the sla comment 2.',  
'Label' => 'Comment2',  
'Module' => 'Kernel::Output::HTML::SLAPreferences::Generic',  
'PrefKey' => 'Comment2',  
'Rows' => '5'  
};
```

Ticket → Frontend::Service::Preferences

ServicePreferences###Comment2

Parameters of the example service attribute Comment2.

Diese Einstellung ist standardmäßig nicht aktiv.

Standardwert:

```
$Self->{'ServicePreferences'}->{'Comment2'} = {  
'Block' => 'TextArea',  
'Cols' => '50',  
'Desc' => 'Define the service comment 2.',  
'Label' => 'Comment2',  
'Module' => 'Kernel::Output::HTML::ServicePreferences::Generic',  
'PrefKey' => 'Comment2',  
'Rows' => '5'  
};
```

Anhang C. GNU Free Documentation License

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non-commercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document, that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters), and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats, suitable for input to text formatters. A copy made in an otherwise Transparent file format, whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include: plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include: PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non-commercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location, containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously, at no char-

ge, 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.

